

**PENGEMBANGAN SISTEM INFORMASI PENGGAJIAN  
KARYAWAN BUS PARIWISATA PT. MAHKOTA MITRA  
SENTOSA MENGGUNAKAN METODE *RATIONAL UNIFIED  
PROCESS* (RUP)**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Tedja Adhi Putra  
NIM: 145150407111055



PROGRAM STUDI SISTEM INFORMASI  
JURUSAN SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018

## PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI PENGGAJIAN KARYAWAN BUS  
PARIWISATA PT MAHKOTA MITRA SENTOSA MENGGUNAKAN METODE  
*RATIONAL UNIFIED PROCESS (RUP)*

### SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Tedja Adhi Putra  
NIM: 145150407111055

Skripsi ini telah diuji dan dinyatakan lulus pada  
27 Juli 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Yusi Tyrone Mursityo, S.Kom., M.AB.  
NIP: 19800228 200604 1 001

Djoko Pramono, S.T., M.Kom.  
NIP: 19780108 200501 1 002

Mengetahui

Ketua Jurusan Sistem Informasi



Dr. Eng., Herman Tolle, S.T., M.T.  
NIP. 19740823 200012 1 001



## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 27 Juli 2018



Tedja Adhi Putra

NIM: 145150407111055

## KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas limpahan rahman yang diberikan sehingga laporan skripsi yang berjudul “Pengembangan Sistem Informasi Penggajian Karyawan Bus Pariwisata PT Mahkota Mitra Sentosa Menggunakan Metode *Rational Unified Process* (RUP)” ini dapat terselesaikan. Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada :

1. Bapak Yusi Tyroni Mursityo, S.Kom.,M.AB, selaku pembimbing satu yang telah mengajarkan ketegasan dan kedisiplinan dalam menyelesaikan kewajiban. Memberikan konsep pemikiran yang luas dalam mencari solusi dari sebuah permasalahan.
2. Bapak Djoko Pramono, S.T., M.Kom., selaku pembimbing dua yang telah mengajarkan ketelitian serta banyak memberikan ilmu, saran, dan motivasi untuk penyelesaian skripsi ini.
3. Suprpto, S.T, M.T selaku Ketua Program Studi Sistem Informasi Jurusan Sistem Informasi Fakultas Ilmu Komputer Universitas Brawijaya
4. Dr. Eng. Herman Tolle, S.T, M.T selaku Ketua Jurusan Sistem Informasi Fakultas Ilmu Komputer Universitas Brawijaya
5. Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya
6. Ayah, ibu, adek beserta seluruh keluarga besar atas segala nasihat, kasih sayang, perhatian, kesabaran serta finansial di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikan skripsi ini.
7. Segenap pimpinan dan karyawan PT Mahkota Mitra Sentosa yang telah mengijinkan serta membantu demi terselesaikan skripsi ini.
8. Keluarga Besar Sistem Informasi yang telah membantu dalam pengerjaan skripsi ini.
9. Seluruh Dosen FILKOM UB yang telah membantu penulis selama menjadi mahasiswa FILKOM UB.

Semoga ALLAH SWT melimpahkan rahmat berlipat ganda kepada seluruh pihak yang telah berkontribusi pada proses pengerjaan skripsi ini. Dengan segala kekurangan, penulis mengharapkan kritik dan saran untuk penelitian selanjutnya. Semoga skripsi ini membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 27 Juli 2018

Penulis

tedjadhi@gmail.com

## ABSTRAK

**Tedja Adhi Putra, Pengembangan Sistem Informasi Penggajian Karyawan Bus Pariwisata PT Mahkota Mitra Sentosa Menggunakan Metode *Rational Unified Process* (RUP)**

**Dosen Pembimbing: Yusi Tyroni Mursityo, S.Kom.,M.AB dan Djoko Pramono, S.T., M.Kom.**

PT Mahkota Mitra Sentosa adalah sebuah perusahaan jasa yang bergerak dalam bidang transportasi darat khususnya bus pariwisata. Pada proses penggajian untuk sopir, kernet dan pembuatan laporan akhir perhitungannya masih menggunakan cara konvensional. Hal ini menimbulkan masalah kurang efisien dalam hal waktu pada proses tersebut. Pada proses perhitungan terkadang harus dilakukan perhitungan ulang untuk memastikan bahwa perhitungan sebelumnya telah benar. Permasalahan yang ada tersebut diperlukan solusi yaitu dengan membangun sistem informasi. Penelitian ini menjelaskan bagaimana membangun sistem informasi yang dibutuhkan oleh PT Mahkota Mitra Sentosa untuk menyelesaikan permasalahan yang ada dengan menggunakan metode pembangunan sistem yaitu *Rational Unified Process*(RUP). Metode tersebut merupakan metodologi pengembangan sistem basis objek dengan tahapan meliputi: *inception*, *elaboration*, *construction*, dan *transition*. Metode ini digunakan karena dapat melakukan proses secara iterasi untuk mengakomodasi perubahan kebutuhan. Hasil penelitian ini adalah sebuah sistem informasi yang berbasis website. Sistem informasi yang telah dibangun diuji menggunakan pengujian *black-box* menghasilkan nilai valid pada semua *test case*. Pengujian *compatibility* digunakan untuk menguji kebutuhan non fungsional dengan hasil terdapat beberapa *minor issue*. *User Acceptance Testing* menghasilkan nilai akhir 87,2%. Pada akhirnya, dapat ditarik kesimpulan bahwa sistem informasi telah dibangun dengan baik dan sesuai dengan kebutuhan.

Kata kunci: *Rational Unified Process*(RUP), sistem informasi, penggajian, bus pariwisata

## ABSTRACT

**Tedja Adhi Putra, Pengembangan Sistem Informasi Penggajian Karyawan Bus Pariwisata PT Mahkota Mitra Sentosa Menggunakan Metode *Rational Unified Process* (RUP)**

**Supervisors: Yusi Tyroni Mursityo, S.Kom.,M.AB and Djoko Pramono, S.T., M.Kom.**

*PT Mahkota Mitra Sentosa is a service company that runs in land transportation especially tour bus. On the payroll process for driver, assistant, and the final report of the calculation is still using conventional method. This causes an issue of time inefficiency in the process. In the calculation process, sometime it is needed to recalculate to make sure that the previous calculation is correct. The problems exist need a solution which is developing an information sistem. This study explains how to develop an information sistem needed by PT Mahkota Mitra Sentosa to resolve the existing problems by using sistem development method which is Rational Unified System. The method is an object-based development sistem by several steps which are: inception, elaboration, construction, and transition. This method is used because it can do the process iteratively to accommodate the change of need. The result of this study is a web-based information sistem. The developed information sistem is tested by using black-box testing generated valid values in all test cases. Compatibility testing is used to test non-functional requirements with the results of several minor issues. User Acceptance Testing generated a final value of 87.2%. In the end, it can be deduced that the information sistem has been built properly and in accordance with the needs. Finally, it is concluded that the information sistem has been well-developed and corresponded the need.*

**Keywords:** *Rational Unified Process(RUP), information sistem, payroll, tour bus*

## DAFTAR ISI

PERNYATAAN ORISINALITAS.....	i
KATA PENGANTAR .....	ii
ABSTRAK .....	iii
<i>ABSTRACT</i> .....	iv
DAFTAR ISI .....	v
DAFTAR TABEL .....	xi
DAFTAR GAMBAR .....	xiv
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan masalah .....	3
1.3 Tujuan .....	3
1.4 Manfaat .....	4
1.5 Batasan masalah .....	4
1.6 Sistematika pembahasan.....	5
BAB 2 LANDASAN KEPUSTAKAAN.....	6
2.1 Kajian Pustaka .....	6
2.2 PT Mahkota Mitra Sentosa .....	11
2.3 Proses Bisnis.....	11
2.3.1 Analisis dan Pemodelan Proses Bisnis <i>As-Is</i> .....	11
2.3.2 Analisis dan Pemodelan Proses Bisnis <i>To-Be</i> .....	11
2.4 <i>Bussiness Process Model dan Notation</i> .....	12
2.5 Sistem .....	15
2.6 Informasi .....	16
2.7 Sistem Informasi.....	16
2.8 Sistem Penggajian .....	16
2.9 <i>Vision Document</i> .....	17
2.9.1 Pemangku Kepentingan .....	17
2.9.2 Tipe Pemangku Kepentingan .....	17
2.9.3 Perwakilan dan Peran Pemangku Kepentingan .....	18
2.9.4 Pernyataan Permasalahan .....	18
2.9.5 Kebutuhan Pemangku Kepentingan .....	19
2.9.6 Kedudukan Produk .....	20



2.9.7 Fitur.....	20
2.10 Basis Data dan DBMS.....	21
2.11 <i>Unified Modeling Language (UML)</i> .....	21
2.11.1 <i>Use-case Diagram</i> .....	21
2.11.2 <i>Class Diagram</i> .....	23
2.11.3 <i>Activity Diagram</i> .....	23
2.11.4 <i>Sequence diagram</i> .....	25
2.12 <i>Rational unified Process (RUP)</i> .....	26
2.13 <i>Model - View – Controller</i> .....	28
2.14 Pengujian .....	29
2.14.1 <i>Black-box Testing</i> .....	29
2.14.2 <i>User Acceptance Testing</i> .....	29
2.14.3 <i>Compatibility</i> .....	30
2.15 Skala Likert .....	31
BAB 3 METODOLOGI PENELITIAN .....	33
3.1 Tempat dan Waktu Penelitian .....	33
3.2 Langkah-Langkah Penelitian .....	33
3.2.1 Studi Literatur .....	34
3.2.2 Pengumpulan Data .....	34
3.2.3 Pemodelan Proses Bisnis .....	34
3.2.4 <i>Business Process Improvement</i> .....	35
3.2.5 Analisis Kebutuhan .....	35
3.2.6 Perancangan Sistem .....	35
3.2.7 Implementasi.....	35
3.2.8 Pengujian .....	36
3.2.9 <i>Deployment</i> .....	36
3.2.10 <i>User Acceptance Testing (UAT)</i> .....	36
3.2.11 Kesimpulan dan Saran .....	37
BAB 4 ANALISIS KEBUTUHAN .....	38
4.1 Pemodelan Proses Bisnis .....	38
4.1.1 Proses Bisnis <i>As-Is</i> .....	38
4.2 <i>Business Process Improvement</i> .....	42
4.2.1 Proses Bisnis <i>To-Be</i> .....	42
4.2.2 Tabel Perbandingan .....	50

4.3 Vision Document .....	50
4.3.1 Identifikasi Tipe Pengguna dan Pemangku Kepentingan .....	50
4.3.2 Identifikasi Perwakilan Pemangku Kepentingan .....	52
4.3.3 Pernyataan Masalah .....	54
4.3.4 Kebutuhan Pemangku Kepentingan .....	55
4.3.5 Kedudukan Produk .....	57
4.3.6 Fitur .....	58
4.4 Kebutuhan Fungsional dan Non Fungsional .....	60
4.4.1 Kebutuhan Fungsional .....	60
4.4.2 Kebutuhan Non Fungsional .....	63
4.5 Pemodelan <i>Use-case</i> .....	63
4.5.1 Deskripsi Aktor .....	64
4.5.2 Skenario <i>Use-case</i> .....	65
4.6 Pemodelan Aktivitas .....	84
4.6.1 <i>Activity Diagram Login</i> .....	84
4.6.2 <i>Activity Diagram Logout</i> .....	85
4.6.3 <i>Activity Diagram</i> Melihat Jadwal Armada Bus .....	85
4.6.4 <i>Activity Diagram</i> Mengelola Armada Bus (Menambah Armada Bus) ..	86
4.6.5 <i>Activity Diagram</i> Mengelola Armada Bus (Menghapus Armada Bus) ..	87
4.6.6 <i>Activity Diagram</i> Mengelola Armada Bus (Menyunting Armada Bus) ..	88
4.6.7 <i>Activity Diagram</i> Mengelola Daftar Tujuan (Menambah Daftar Tujuan)	89
4.6.8 <i>Activity Diagram</i> Mengelola Daftar Tujuan (Menghapus Daftar Tujuan)	90
4.6.9 <i>Activity Diagram</i> Mengelola Daftar Tujuan (Menyunting Daftar Tujuan)	91
4.6.10 <i>Activity Diagram</i> Mengelola Persewaan (Menambah Persewaan) ...	92
4.6.11 <i>Activity Diagram</i> Mengelola Persewaan (Menghapus Persewaan) ...	93
4.6.12 <i>Activity Diagram</i> Mengelola Persewaan (Menyunting Persewaan) ..	94
4.6.13 <i>Activity Diagram</i> Mencetak Laporan Akhir Bulan per Unit Bus .....	95
4.6.14 <i>Activity Diagram</i> Mencetak Laporan Akhir Bulan Global .....	96
4.6.15 <i>Activity Diagram</i> Mencetak Slip Gaji .....	97
4.6.16 <i>Activity Diagram</i> Mengelola Pegawai (Menambah Pegawai) .....	98
4.6.17 <i>Activity Diagram</i> Mengelola Pegawai (Menghapus Pegawai) .....	99
4.6.18 <i>Activity Diagram</i> Mengelola Pegawai (Menyunting Pegawai) .....	100

4.6.19 <i>Activity Diagram</i> Mengelola Pengguna Sistem (Menambah Pengguna Sistem) .....	101
4.6.20 <i>Activity Diagram</i> Mengelola Pengguna Sistem (Menghapus Pengguna Sistem) .....	103
4.6.21 <i>Activity Diagram</i> Mengelola Pengguna Sistem (Menyunting Pengguna Sistem) .....	104
4.6.22 <i>Activity Diagram</i> Menyunting Profil .....	105
BAB 5 PERANCANGAN .....	106
5.1 <i>Sequence diagram</i> .....	106
5.1.1 Melihat Jadwal Bus .....	106
5.1.2 Mengelola Persewaan .....	106
5.1.3 Cetak Slip Gaji .....	109
4.2 <i>Class Diagram</i> .....	110
4.3 Pemodelan Data .....	112
4.3.1 Tabel Armada_bus .....	113
4.3.2 Tabel Daftar_tujuan .....	114
4.3.3 Tabel Gaji .....	114
4.3.4 Tabel Jenis_pekerjaan .....	115
4.3.5 Tabel Operasioinal_bus .....	115
4.3.6 Tabel Operasional_tambahan .....	116
4.3.7 Tabel Pegawai .....	117
4.3.8 Tabel Pengguna_sistem .....	117
4.3.9 Tabel Persewaan .....	119
4.4 Antarmuka Pengguna .....	119
4.4.1 Antarmuka <i>Login</i> .....	119
4.4.2 Antarmuka <i>Home</i> .....	120
4.4.3 Antarmuka Jadwal Bus .....	121
4.4.4 Antarmuka Persewaan Bus .....	122
4.4.5 Antarmuka Persewaan Bus : Menambah Persewaan .....	123
4.4.6 Antarmuka Persewaan Bus : Menyunting Persewaan .....	123
4.4.7 Antarmuka Laporan Unit .....	124
4.4.8 Antarmuka Laporan Global .....	125
4.4.9 Antarmuka Gaji .....	126
4.4.10 Antarmuka Gaji : Cetak Slip Gaji .....	126
4.5 Perancangan Algoritme .....	127

4.5.1 Algoritme fungsi displayJadwalBus()	127
4.5.2 Algoritme fungsi getJadwalBus()	127
4.5.3 Algoritme fungsi displayPersewaanBus()	128
4.5.4 Algoritme fungsi addPersewaanBus()	128
4.5.5 Algoritme fungsi updatePersewaanBus()	129
4.5.6 Algoritme fungsi getLaporanUnit()	129
4.5.7 Algoritme fungsi getLaporanGlobal()	129
4.5.8 Algoritme fungsi getStatusGaji()	130
4.5.9 Algoritme fungsi displayGajiPrint()	130
<b>BAB 6 IMPLEMENTASI</b>	<b>132</b>
6.1 Spesifikasi Lingkungan Implementasi	132
6.2 Implementasi Algoritme	133
6.2.1 Algoritme fungsi displayJadwalBus()	133
6.2.2 Algoritme fungsi getJadwalBus()	133
6.2.3 Algoritme fungsi displayPersewaanBus()	134
6.2.4 Algoritme fungsi addPersewaanBus()	134
6.2.5 Algoritme fungsi updatePersewaanBus()	135
6.2.6 Algoritme fungsi getLaporanUnit()	136
6.2.7 Algoritme fungsi getLaporanGlobal()	137
6.2.8 Algoritme fungsi getStatusGaji()	138
6.2.9 Algoritme fungsi displayGajiPrint()	138
6.3 Implementasi Antarmuka Pengguna	139
6.3.1 Antarmuka <i>Login</i>	140
6.3.2 Antarmuka <i>Home</i>	140
6.3.3 Antarmuka Jadwal Bus	141
6.3.4 Antarmuka Persewaan Bus	141
6.3.5 Antarmuka Persewaan Bus: Menambah Persewaan	142
6.3.6 Antarmuka Persewaan Bus: Menyunting Persewaan	143
6.3.7 Antarmuka Laporan Unit	144
6.3.8 Antarmuka Laporan Global	145
6.3.9 Antarmuka Gaji	145
6.3.10 Antarmuka Gaji: Slip Gaji	146
6.4 <i>Deployment</i>	147
<b>BAB 7 PENGUJIAN</b>	<b>148</b>

7.1 Pengujian <i>Black-Box</i> .....	148
7.1.1 Pengujian <i>Black-Box</i> Melakukan <i>Login</i> .....	148
7.1.2 Pengujian <i>Black-Box</i> Melakukan <i>Logout</i> .....	150
7.1.3 Pengujian <i>Black-Box</i> Melihat Jadwal Armada Bus .....	151
7.1.4 Pengujian <i>Black-Box</i> Mengelola Armada Bus .....	151
7.1.5 Pengujian <i>Black-Box</i> Mengelola Persewaan Bus .....	155
7.1.6 Pengujian <i>Black-Box</i> Mencetak Laporan Unit.....	157
7.1.7 Pengujian <i>Black-Box</i> Mencetak Laporan Global .....	158
7.1.8 Pengujian <i>Black-Box</i> Mencetak Slip Gaji.....	159
7.1.9 Pengujian <i>Black-Box</i> Mengelola Pegawai .....	161
7.1.10 Pengujian <i>Black-Box</i> Mengelola Pengguna Sistem .....	164
7.1.11 Pengujian <i>Black-Box</i> Menyunting Profil .....	168
7.1.12 Pengujian <i>Black-Box</i> Mengelola Daftar Tujuan.....	169
7.2 Pengujian <i>Compatibility</i> .....	173
7.3 Pengujian <i>User Acceptance Testing (UAT)</i> .....	173
BAB 8 KESIMPULAN DAN SARAN.....	181
8.1 Kesimpulan .....	181
8.2 Saran.....	182
DAFTAR PUSTAKA .....	183
LAMPIRAN A HASIL WAWANCARA .....	185
LAMPIRAN B USER ACCEPTANCE TESTING.....	187



## DAFTAR TABEL

Tabel 2.1 Elemen pada BPMN (Object Mangement Group, Inc. 2011) .....	12
Tabel 2.2 Pernyataan Masalah yang Telah Diadaptasi (Bittner & Spence, 2002) .	18
Tabel 2.3 Simbol pada <i>activity diagram</i> .....	23
Tabel 2.4 Bobot Nilai .....	31
Tabel 2.5 Persentase nilai .....	31
Tabel 4.1 Perubahan Aktivitas Proses Bisnis Sewa Bus .....	45
Tabel 4.2 Proses Bisnis Membuat Slip Gaji .....	46
Tabel 4.3 Proses Bisnis Membuat Laporan Unit .....	48
Tabel 4.4 Proses Bisnis Membuat Laporan Global .....	49
Tabel 4.5 Perbandingan Waktu Pelaksanaan Proses Bisnis .....	50
Tabel 4.6 Tabel Tipe Pengguna dan Pemangku Kepentingan .....	51
Tabel 4.7 Tabel Informasi Pengguna .....	51
Tabel 4.8 Tabel Identifikasi Perwakilan Pemangku Kepentingan .....	53
Tabel 4.9 Tabel Pernyataan Masalah .....	54
Tabel 4.10 Tabel Kebutuhan Pemangku Kepentingan .....	55
Tabel 4.11 Tabel Kedudukan Produk .....	58
Tabel 4.12 Tabel Fitur .....	59
Tabel 4.13 Tabel Kebutuhan Fungsional .....	60
Tabel 4.14 Tabel Kebutuhan Non Fungsional .....	63
Tabel 4.15 Tabel Deskripsi Aktor .....	64
Tabel 4.16 Skenario <i>Use-Case</i> Melakukan <i>Login</i> .....	65
Tabel 4.17 Skenario <i>Use-Case</i> Melakukan <i>Logout</i> .....	66
Tabel 4.18 Skenario <i>Use-Case</i> Melihat Jadwal Armada Bus .....	67
Tabel 4.19 Skenario <i>Use-Case</i> Mengelola Armada Bus .....	68
Tabel 4.20 Skenario <i>Use-Case</i> Mengelola Daftar Tujuan .....	70
Tabel 4.21 Skenario <i>Use-Case</i> Mengelola Persewaan .....	72
Tabel 4.22 <i>Use-Case</i> Skenario Mencetak Laporan Akhir Bulan per Unit Bus .....	74
Tabel 4.23 Skenario <i>Use-Case</i> Mencetak Laporan Akhir Bulan Global .....	76
Tabel 4.24 Skenario <i>Use-Case</i> Mencetak Slip Gaji .....	77
Tabel 4.25 <i>Use-Case</i> Skenario Mengelola Pegawai .....	78
Tabel 4.26 <i>Use-Case</i> Skenario Mengelola Pengguna Sistem .....	80
Tabel 4.27 <i>Use-Case</i> Skenario Menyunting Profil .....	83
Tabel 5.1 Tabel Armada_bus .....	113
Tabel 5.2 Tabel Daftar_tujuan .....	114
Tabel 5.3 Tabel Gaji .....	115
Tabel 5.4 Tabel Jenis_pekerjaan .....	115
Tabel 5.5 Tabel Operasional_bus .....	115
Tabel 5.6 Tabel Operasional_tambahan .....	116
Tabel 5.7 Tabel Pegawai .....	117
Tabel 5.8 Tabel Pengguna_sistem .....	117
Tabel 5.9 Tabel Persewaan .....	119
Tabel 5.10 Algoritme fungsi displayJadwalBus() .....	127
Tabel 5.11 Algoritme fungsi getJadwalBus() .....	127
Tabel 5.12 Algoritme fungsi displayPersewaanBus() .....	128

Tabel 5.13 Algoritme fungsi addPersewaanBus()	128
Tabel 5.14 Algoritme fungsi updatePersewaanBus()	129
Tabel 5.15 Algoritme fungsi getLaporanUnit()	129
Tabel 5.16 Algoritme fungsi getLaporanGlobal()	130
Tabel 5.17 Algoritme fungsi getStatusGaji()	130
Tabel 5.18 Algoritme fungsi displayGajiPrint()	131
Tabel 6.1 Spesifikasi Perangkat Keras	132
Tabel 6.2 Spesifikasi Perangkat Lunak	132
Tabel 6.3 Implementasi Algoritme fungsi displayJadwalBus()	133
Tabel 6.4 Implementasi Algoritme fungsi getJadwalBus()	133
Tabel 6.5 Implementasi Algoritme fungsi displayPersewaanBus()	134
Tabel 6.6 Implementasi Algoritme fungsi addPersewaanBus()	134
Tabel 6.7 Implementasi Algoritme fungsi updatePersewaanBus()	136
Tabel 6.8 Implementasi Algoritme fungsi getLaporanUnit()	136
Tabel 6.9 Implementasi Algoritme fungsi getLaporanGlobal()	137
Tabel 6.10 Implementasi Algoritme fungsi getStatusGaji()	138
Tabel 6.11 Implementasi Algoritme fungsi displayGajiPrint()	138
Tabel 7.1 Pengujian <i>Black-Box</i> Melakukan <i>Login</i>	148
Tabel 7.2 Kasus Uji Validasi Melakukan <i>Login</i>	148
Tabel 7.3 Kasus Uji Validasi Melakukan <i>Login</i> : A1 <i>Unidentified User</i>	149
Tabel 7.4 Pengujian <i>Black-Box</i> Melakukan <i>Logout</i>	150
Tabel 7.5 Kasus Uji Validasi Melakukan <i>Logout</i>	150
Tabel 7.6 Pengujian <i>Black-Box</i> Melihat Jadwal Armada Bus	151
Tabel 7.7 Kasus Uji Validasi Melihat Jadwal Armada Bus	151
Tabel 7.8 Pengujian <i>Black-Box</i> Mengelola Armada Bus	152
Tabel 7.9 Kasus Uji Validasi Mengelola Armada Bus	152
Tabel 7.10 Kasus Uji Validasi Mengelola Armada Bus: A1 Menghapus Armada Bus	153
Tabel 7.11 Kasus Uji Validasi Mengelola Armada Bus: A2 Menyunting Armada Bus	153
Tabel 7.12 Kasus Uji Validasi Mengelola Armada Bus: A3 <i>Incomplete Form</i>	154
Tabel 7.13 Pengujian <i>Black-Box</i> Mengelola Persewaan Bus	155
Tabel 7.14 Kasus Uji Validasi Mengelola Persewaan	155
Tabel 7.15 Kasus Uji Validasi Mengelola Persewaan: A1 Menyunting Persewaan Bus	156
Tabel 7.16 Kasus Uji Validasi Mengelola Persewaan: A2 <i>Incomplete Form</i>	157
Tabel 7.17 Pengujian <i>Black-Box</i> Mencetak Laporan Unit	157
Tabel 7.18 Kasus Uji Validasi Mencetak Laporan Unit	158
Tabel 7.19 Pengujian <i>Black-Box</i> Mencetak Laporan Global	158
Tabel 7.20 Kasus Uji Validasi Mencetak Laporan Global	159
Tabel 7.21 Pengujian <i>Black-Box</i> Mencetak Slip Gaji	159
Tabel 7.22 Kasus Uji Validasi Mencetak Slip Gaji	160
Tabel 7.23 Kasus Uji Validasi Mencetak Slip Gaji: A1 Data tidak ditemukan	160
Tabel 7.24 Pengujian <i>Black-Box</i> Mengelola Pegawai	161
Tabel 7.25 Kasus Uji Validasi Mengelola Pegawai	162

Tabel 7.26 Kasus Uji Validasi Mengelola Pegawai: A1 Menghapus Pegawai .....	162
Tabel 7.27 Kasus Uji Validasi Mengelola Pegawai: A2 Menyunting Pegawai .....	163
Tabel 7.28 Kasus Uji Validasi Mengelola Pegawai: A3 <i>Incomplete Form</i> .....	164
Tabel 7.29 Pengujian <i>Black-Box</i> Mengelola Pengguna Sistem .....	164
Tabel 7.30 Kasus Uji Validasi Mengelola Pengguna Sistem .....	165
Tabel 7.31 Kasus Uji Validasi Mengelola Pengguna Sistem: A1 Menghapus Pengguna Sistem .....	166
Tabel 7.32 Kasus Uji Validasi Mengelola Pengguna Sistem: A2 Menyunting Pengguna Sistem .....	166
Tabel 7.33 Kasus Uji Validasi Mengelola Pengguna Sistem: A3 <i>Incomplete Form</i> .....	167
Tabel 7.34 Pengujian <i>Black-Box</i> Menyunting Profil .....	168
Tabel 7.35 Kasus Uji Validasi Menyunting Profil .....	168
Tabel 7.36 Kasus Uji Validasi Menyunting Profil: A1 <i>Incomplete Form</i> .....	169
Tabel 7.37 Pengujian <i>Black-Box</i> Mengelola Daftar Tujuan .....	169
Tabel 7.38 Kasus Uji Validasi Mengelola Daftar Tujuan .....	170
Tabel 7.39 Kasus Uji Validasi Mengelola Daftar Tujuan: A1 Menghapus Daftar Tujuan .....	171
Tabel 7.40 Kasus Uji Validasi Mengelola Daftar Tujuan: A2 Menyunting Daftar Tujuan .....	171
Tabel 7.41 Kasus Uji Validasi Mengelola Pengguna Sistem: A3 <i>Incomplete Form</i> .....	172
Tabel 7.42 Analisa <i>User Acceptance Testing</i> .....	174
Tabel 7.43 Kriteria dan Pernyataan <i>User Acceptance Testing</i> .....	175

## DAFTAR GAMBAR

Gambar 2.1 Contoh BPMN pada proses bisnis aplikasi dokter .....	12
Gambar 2.2 Notasi Aktor .....	22
Gambar 2.3 Notasi <i>Use-Case</i> .....	22
Gambar 2.4 <i>Use-Case</i> Diagram pada sistem musik .....	22
Gambar 2.5 <i>Class Diagram</i> kelas <i>Thoroughbred</i> .....	23
Gambar 2.6 <i>Activity Diagram</i> .....	24
Gambar 2.7 Elemen-elemen pada <i>Sequence diagram</i> .....	25
Gambar 2.8 Contoh <i>Sequence diagram</i> .....	26
Gambar 2.9 Fase pada Metode RUP .....	26
Gambar 2.10 Arsitektur berpola MVC .....	28
Gambar 2.11 Contoh Skala Likert .....	31
Gambar 3.3.1 Alur Penelitian .....	33
Gambar 4.1 Proses Bisnis Sewa Bus .....	40
Gambar 4.2 Proses Bisnis Membuat Slip Gaji .....	41
Gambar 4.3 Proses Bisnis Membuat Laporan Unit .....	41
Gambar 4.4 Proses Bisnis Membuat Laporan Global .....	42
Gambar 4.5 Proses Bisnis Sewa Bus .....	44
Gambar 4.6 Proses Bisnis Membuat Slip Gaji .....	46
Gambar 4.7 Proses Bisnis Membuat Laporan Unit .....	47
Gambar 4.8 Proses Bisnis Membuat Laporan Global .....	49
Gambar 4.9 Diagram <i>Use-case</i> .....	64
Gambar 4.10 <i>Activity Diagram Login</i> .....	84
Gambar 4.11 <i>Activity Diagram Logout</i> .....	85
Gambar 4.12 <i>Activity Diagram</i> Melihat Jadwal Armada .....	86
Gambar 4.13 <i>Activity Diagram</i> Menambah Armada Bus .....	87
Gambar 4.14 <i>Activity Diagram</i> Menghapus Armada Bus .....	88
Gambar 4.15 <i>Activity Diagram</i> Menyunting Armada Bus .....	89
Gambar 4.16 <i>Activity Diagram</i> Menambah Daftar Tujuan .....	90
Gambar 4.17 <i>Activity Diagram</i> Menghapus Daftar Tujuan .....	91
Gambar 4.18 <i>Activity Diagram</i> Menyunting Daftar Tujuan .....	92
Gambar 4.19 <i>Activity Diagram</i> Menambah Persewaan .....	93
Gambar 4.20 <i>Activity Diagram</i> Menghapus Persewaan .....	94
Gambar 4.21 <i>Activity Diagram</i> Menyunting Persewaan .....	95
Gambar 4.22 <i>Activity Diagram</i> Mencetak Laporan Akhir Bulan per Unit .....	96
Gambar 4.23 <i>Activity Diagram</i> Laporan Akhir Bulan Global .....	97
Gambar 4.24 <i>Activity Diagram</i> Mencetak Slip Gaji .....	98
Gambar 4.25 <i>Activity Diagram</i> Menambah Pegawai .....	99
Gambar 4.26 <i>Activity Diagram</i> Menghapus Pegawai .....	100
Gambar 4.27 <i>Activity Diagram</i> Menyunting Pegawai .....	101
Gambar 4.28 <i>Activity Diagram</i> Menambah Pengguna Sistem .....	102
Gambar 4.29 <i>Activity Diagram</i> Menghapus Pengguna Sistem .....	103
Gambar 4.30 <i>Activity Diagram</i> Menyunting Pengguna Sistem .....	104
Gambar 4.31 <i>Activity Diagram</i> Menyunting Profil .....	105
Gambar 5.1 <i>Sequence diagram</i> Melihat Jadwal Bus .....	106

Gambar 5.2 <i>Sequence Diagram</i> Mengelola Persewaan .....	107
Gambar 5.3 <i>Sequence Diagram</i> Mengelola Persewaan: Menambah Persewaan .....	108
Gambar 5.4 <i>Sequence Diagram</i> Mengelola Persewaan: Menyunting Persewaan .....	109
Gambar 5.5 <i>Sequence Diagram</i> Cetak Slip Gaji .....	110
Gambar 5.6 <i>Class Diagram</i> sebagai Logical Class .....	111
Gambar 5.7 <i>Class Diagram</i> sebagai Domain Model .....	112
Gambar 5.8 <i>Physical Data Modelling</i> .....	113
Gambar 5.9 Antarmuka <i>Login</i> .....	120
Gambar 5.10 Antarmuka Home .....	120
Gambar 5.11 Antarmuka Jadwal Bus .....	121
Gambar 5.12 Antarmuka Jadwal Bus .....	122
Gambar 5.13 Antarmuka Persewaan Bus .....	122
Gambar 5.14 Antarmuka Persewaan Bus : Menambah Persewaan .....	123
Gambar 5.15 Antarmuka Persewaan Bus : Menyunting Persewaan .....	123
Gambar 5.16 Antarmuka Laporan Unit .....	124
Gambar 5.17 Antarmuka Laporan Unit (hasil iterasi) .....	124
Gambar 5.18 Antarmuka Laporan Global .....	125
Gambar 5.19 Antarmuka Laporan Global (hasil iterasi) .....	125
Gambar 5.20 Antarmuka Gaji .....	126
Gambar 5.21 Antarmuka Gaji : Cetak Slip Gaji .....	126
Gambar 6.1 Implementasi Antarmuka <i>Login</i> .....	140
Gambar 6.2 Implementasi Antarmuka Home .....	141
Gambar 6.3 Implementasi Antarmuka Jadwal Bus .....	141
Gambar 6.4 Implementasi Antarmuka Persewaan Bus .....	142
Gambar 6.5 Implementasi Antarmuka Menambah Persewaan .....	143
Gambar 6.6 Implementasi Antarmuka Menyunting Persewaan .....	144
Gambar 6.7 Implementasi Antarmuka Laporan Unit .....	145
Gambar 6.8 Implementasi Antarmuka Laporan Global .....	145
Gambar 6.9 Implementasi Antarmuka Gaji .....	146
Gambar 6.10 Implementasi Antarmuka Slip Gaji .....	147
Gambar 7.1 Kodifikasi Pengujian <i>Black-Box</i> .....	148
Gambar 7.2 Kategori <i>compatibility issues</i> SortSite .....	173
Gambar 7.3 Hasil Pengujian SortSite .....	173





# TEDJA ADHI PUTRA

## CURRICULUM VITAE

INFORMATION SYSTEM LABORATORY GRADUATE

### NOMOR INDUK MAHASISWA

145150407111055

### TEMPAT, TANGGAL LAHIR

Kediri, 20 November 1995

### ALAMAT

Jl. Raya Pojok RT 01 RW 03 Dsn. Pojok, Ds. Plemahan  
Kec. Plemahan, Kab. Kediri, 64155



### RIWAYAT PENDIDIKAN

2001	2002	TK Darma Wanita
2002	2008	SDN Plemahan 1
2008	2011	SMPN 1 Plemahan
2011	2014	SMAN 2 Pare
2014	2018	Universitas Brawijaya



### KEMAMPUAN

HTML, PHP, Java, JavaScript  
MySQL, DB2, SQL Server, Oracle  
Microsoft Office Word, Excel,  
PowerPoint



### PENGALAMAN PRIBADI

2012	2013	Wakil Ketua Pramuka SMAN 2 Pare
2014	2015	Bendahara Karang Taruna Pojok
2015	2016	Humas Ikatan Alumni SMAN 2 PARE
2016	2017	Asisten Praktikum Sistem Basis Data Laboratorium SI 2016
2016	2017	Asisten Praktikum Dasar Basis Data Laboratorium SI 2017
2017	2018	Asisten Praktikum Sistem Basis Data Laboratorium SI 2017



### BAHASA

Indonesia (Bahasa Pokok)  
Inggris



### PASSION

Music  
Traveling



### KONTAK

+62 856 4982 0165  
tedjadhi@gmail.com  
tedjadhi

## BAB 1 PENDAHULUAN

### 1.1 Latar Belakang

Sarana transportasi telah menjadi bagian yang penting dari kehidupan manusia pada zaman sekarang ini. Hal ini dikarenakan transportasi merupakan salah satu sarana utama bagi manusia dalam kehidupan sehari-hari untuk bergerak dan berpindah dari satu tempat ke tempat yang lainnya dengan cepat, serta diikuti dengan perkembangan teknologi yang begitu pesat. Kedua hal tersebut sangat berdampak pada persaingan usaha dalam meningkatkan kinerja perusahaannya. Dalam hal ini penggunaan teknologi dan informasi merupakan hal yang sangat penting dalam menjalankan suatu pekerjaan dan kegiatan. Sehingga dapat memengaruhi aspek kehidupan dari segi perkembangan maupun segi ekonomis perusahaan di masa yang akan datang.

Dalam perusahaan pasti mempunyai beberapa pegawai. Dengan adanya pegawai, visi misi suatu perusahaan tersebut akan dapat dicapai. Dalam mencapai visi misi yang telah direncanakan, perusahaan harus menggunakan sumber daya manusia yang terampil dan cerdas di bidangnya masing-masing. Dengan begitu pimpinan perusahaan harus bisa merangsang karyawannya untuk dapat melakukan tugas yang diberikan dengan baik. Salah satu cara dalam merangsang kerja karyawannya yaitu dengan memberikan gaji.

Gaji merupakan sejumlah uang yang diberikan kepada pegawai sebagai imbalan atas jasa atau pekerjaan yang telah dilakukan terhadap perusahaan. Setiap perusahaan memiliki sistem yang berbeda dalam memberikan gaji. Gaji yang diberikan kepada karyawan juga berbeda sesuai dengan jabatan dan tingkat golongannya. Sehingga apabila suatu perusahaan mengalami kesulitan dalam perhitungan penggajian bukanlah hal yang mengherankan. Hal ini terjadi karena adanya jumlah tenaga kerja yang banyak dan waktu yang digunakan untuk menghitung gaji juga singkat biasanya dilakukan di akhir bulan.

Perkembangan teknologi di Indonesia yang telah maju pesat. Salah satunya adalah perkembangan teknologi informasi, di mana informasi yang cepat dan tepat dibutuhkan. Menggunakan cara konvensional untuk mencari informasi secara cepat dan tepat tidaklah mudah karena membutuhkan waktu untuk mencari datanya juga terkadang bisa terjadi kesalahan yang disebabkan oleh kelalaian dari sumber daya manusianya. Maka dari itu dibutuhkan suatu sistem yang sudah terkomputerisasi, dengan cara mengubah semua sistem yang masih konvensional ke sistem yang terkomputerisasi. Dengan menggunakan sistem terkomputerisasi akan lebih efisien, ini dapat dilihat dari proses kecepatan, ketepatan, biaya, dan sebagainya. Mulai dari yang pertama yaitu kecepatan, dengan sistem terkomputerisasi semua data akan tersimpan dalam komputer sehingga akan mempermudah dalam pencarian data yang dibutuhkan daripada mencari pada data yang tersimpan secara konvensional. Untuk yang kedua mengenai ketepatan, tentu saja sistem yang terkomputerisasi jauh lebih tepat dibandingkan sistem yang konvensional. Sedangkan masalah biaya pada sistem

yang terkomputerisasi biayanya lebih hemat dibandingkan yang konvensional, penghematannya dapat dilihat pada penggunaan kertas-kertas yang terlalu berlebihan, serta tempat penyimpanan berkas-berkas.

PT Mahkota Mitra Sentosa adalah sebuah perusahaan jasa yang bergerak dalam bidang transportasi darat khususnya bus pariwisata. Jumlah armada yang ada di PT Mahkota Mitra Sentosa ada 25 armada. Jumlah sopir dan kernet yaitu 55 orang termasuk cadangan. Dalam pengolahan data penyewaan bus pariwisata dan pemberian komisi *crew* bus di PT Mahkota Mitra Sentosa masih menggunakan sistem kerja konvensional. Misalnya saat mencatat data orang yang menyewa bus, pegawai mencatat identitas orang tersebut di dalam buku besar. Perhitungan komisi *crew* bus yang terjadi saat ini masih menggunakan kalkulator. Sehingga pada proses perhitungan tersebut terkadang pegawai harus melakukan perhitungan ulang untuk memastikan bahwa perhitungan sebelumnya telah benar. Kemudian pegawai memberikan komisi sesuai dengan perhitungan yang didapatkan kepada *crew* bus. Pegawai dalam membuat laporan akhir dilakukan dengan cara menyalin data laporan akhir dari buku besar ke *Microsoft Excel* untuk dicetak. Proses pembuatan laporan tersebut membutuhkan waktu yang tidak sedikit. Kurang efisiennya waktu dalam pendataan dan pengolahan data untuk pencarian data penjadwalan bus, data persewaan, perhitungan komisi, dan dalam pembuatan laporan akhir bulan akan memengaruhi kinerja dari operasional PT Mahkota Mitra Sentosa. Perekapan data yang masih menggunakan cara konvensional memungkinkan dokumen akan rusak atau hilang karena *human error* serta ketika pimpinan bus ingin mengetahui detail dari perekapan data harus datang ke kantor sehingga menimbulkan ketidakefisienan di mana seharusnya ketidakefisienan tersebut dapat diminimalisir dengan bantuan teknologi informasi.

Salah satu penelitian yang dilakukan oleh Sri Rahayu (2017) berjudul "Perancangan Aplikasi Penggajian Berbasis *Web* di Sekolah Menengah Kejuruan Islam Atturmudziyyah Garut" menghasilkan aplikasi penggajian yang memiliki fasilitas untuk pengolahan data gaji guru atau pegawai dilakukan secara otomatis. Bahkan mempermudah bagi guru atau pegawai dalam melihat data gaji dan mencetak slip gaji agar tidak mengantri sehingga efisien dalam hal waktu. Pada penelitian ini peneliti menggunakan metode pengembangan *Rational Unified Process (RUP)* dalam proses pengembangan perangkat lunaknya. Pemodelan yang digunakan yaitu *Unified Modeling Language* dengan diagram yang digunakan *use-case diagram*, *activity diagram*, dan *class diagram*.

Berdasarkan penelitian yang telah dilakukan, penulis menyimpulkan bahwa untuk mengimplementasikan sebuah perekapan data dan penggajian yang baik perlunya menggunakan sebuah sistem informasi sehingga dari data yang ada dapat diolah secara cepat dan akurat. Serta dapat menghemat waktu yang digunakan untuk melakukan proses pengolahan data. Begitupun dengan metode pengembangan sistem yang digunakan, menggunakan metode pengembangan sistem *Rational Unified Process (RUP)*. *Rational Unified Process (RUP)* merupakan metode pengembangan sistem yang menggunakan *use-case driven* dan

pendekatan iteratif. Menggunakan *Rational Unified Process (RUP)* dapat membantu dalam proses pengembangan perangkat lunak PT Mahkota Mitra Sentosa karena dapat melakukan proses iterasi. Sehingga apabila dalam proses pengembangan perangkat lunak PT Mahkota Mitra Sentosa membutuhkan perubahan baik dalam penambahan atau pengurangan fitur dapat diakomodasi. Perubahan yang terjadi dalam proses pengembangan tersebut bertujuan agar dapat menghasilkan sebuah perangkat lunak yang berkualitas dan sesuai dengan kebutuhan PT Mahkota Mitra Sentosa. Metode ini pada proses pengembangan sistem dapat terdokumentasi dengan jelas menggunakan UML, sehingga akan mudah untuk dipahami oleh pengembang lain apabila dilakukan pengembangan lanjut terhadap sistem yang telah dibuat.

Dari permasalahan yang telah dijabarkan di atas pada PT Mahkota Mitra Sentosa, maka perlu untuk mengimplementasi teknologi informasi dalam menyelesaikan permasalahan yang terjadi. Teknologi informasi yang akan di implementasi yaitu sistem informasi berbasis *website*. Sistem informasi tersebut dapat mengelola data penyewaan bus dan komisi untuk *crew*. Dengan begitu penulis tertarik ingin melakukan penelitian yang berjudul “PENGEMBANGAN SISTEM INFORMASI PENGGAJIAN KARYAWAN BUS PARIWISATA PT MAHKOTA MITRA SENTOSA MENGGUNAKAN METODE *RATIONAL UNIFIED PROCESS (RUP)*”. Penelitian ini bertujuan untuk memudahkan pimpinan dan karyawan PT Mahkota Mitra Sentosa dalam melakukan proses operasionalnya.

## 1.2 Rumusan masalah

Berdasarkan penjabaran masalah pada latar belakang, dapat dirumuskan permasalahan dalam penelitian adalah sebagai berikut:

1. Bagaimana menganalisis dan memodelkan proses bisnis yang terdapat pada PT Mahkota Mitra Sentosa saat ini dalam melakukan penggajian karyawan dan mengusulkan pemodelan proses bisnis yang baru sesuai dengan sistem informasi yang akan dibuat pada fase *inception*?
2. Bagaimana hasil analisis kebutuhan dan perancangan pengembangan sistem informasi penggajian karyawan bus pariwisata pada PT Mahkota Mitra Sentosa menggunakan metode *Rational Unified Process (RUP)* pada fase *elaboration*?
3. Bagaimana mengimplementasikan dan pengujian sistem pada pengembangan sistem informasi penggajian karyawan bus pariwisata pada PT Mahkota Mitra Sentosa menggunakan metode *Rational Unified Process (RUP)* pada fase *construction* ?
4. Bagaimana hasil pengujian *user acceptance testing* pada pengembangan sistem informasi penggajian karyawan bus pariwisata pada PT Mahkota Mitra Sentosa menggunakan metode *Rational Unified Process (RUP)* pada fase *transtion*?

## 1.3 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut:



1. Melakukan analisis dan memodelkan proses bisnis yang terdapat pada PT MAHKOTA MITRA SENTOSA saat ini dalam melakukan penggajian karyawan dan mengusulkan pemodelan proses bisnis yang baru sesuai dengan sistem informasi yang akan dibuat.
2. Melakukan analisis kebutuhan dan perancangan pengembangan sistem informasi penggajian karyawan bus pariwisata pada PT MAHKOTA MITRA SENTOSA pada fase *elaboration*.
3. Melakukan implementasi dan pengujian sistem pada pengembangan sistem informasi penggajian karyawan bus pariwisata pada PT MAHKOTA MITRA SENTOSA berdasarkan perancangan yang telah dibuat menggunakan metode *Rational Unified Process* (RUP) pada fase *construction*.
4. Melakukan pengujian *user acceptance testing* pada pengembangan sistem informasi penggajian karyawan bus pariwisata pada PT MAHKOTA MITRA SENTOSA menggunakan metode *Rational Unified Process* (RUP) pada fase *transition*.

#### 1.4 Manfaat

Adapun manfaat yang didapat dari penelitian dan penyusunan skripsi ini adalah sebagai berikut:

Bagi penulis

1. Menerapkan teori yang telah didapat pada saat perkuliahan dengan masalah sebenarnya.
2. Memberikan gambaran secara umum tentang pengembangan sistem informasi penggajian karyawan bus pariwisata PT MAHKOTA MITRA SENTOSA menggunakan metode *Rational Unified Process* (RUP).

Bagi PT MAHKOTA MITRA SENTOSA

Diharapkan dengan adanya sistem informasi tersebut dapat membantu dalam proses operasionalnya.

#### 1.5 Batasan masalah

Agar pembahasan permasalahan tidak menyimpang dari pokok bahasan, maka perlu adanya batasan masalah dalam pembuatan sistem informasi ini diantaranya:

1. Merancang dan membangun sistem informasi penggajian karyawan berbasis web pada PT MAHKOTA MITRA SENTOSA.
2. Membangun sistem informasi persewaan berbasis web menggunakan metode berorientasi objek dengan model pengembangan *Rational Unified Process* (RUP).
3. Sistem informasi yang dibangun meliputi perekapan data persewaan bus dan data gaji karyawan.
4. Penggajian karyawan berfokus pada karyawan yang bertugas sebagai sopir dan kernet.



5. Sistem informasi yang dibangun menggunakan bahasa pemrograman *Hypertext Preprocessor* (PHP), *framework CodeIgniter* (CI), dan Mysql sebagai database.

## 1.6 Sistematika pembahasan

Sistematika pembahasan ini menggunakan kerangka penulisan yang tersusun sebagai berikut :

### BAB I Pendahuluan

Bab ini berisi tentang latar belakang masalah, rumusan masalah, dan pokok bahasan, batasan masalah, tujuan, manfaat dari penelitian serta sistematika pembahasan skripsi.

### BAB II Landasan Kepustakaan

Bab ini berisi tinjauan pustaka dari beberapa penelitian yang telah dilakukan sebelumnya menggunakan metode RUP, serta menjelaskan beberapa dasar teori dan referensi yang mendasari perancangan dan implementasi sistem informasi menggunakan metode RUP.

### BAB III Metodologi

Dalam bab ini menjabarkan proses-proses perancangan implementasi sistem informasi dengan menggunakan metode RUP.

### BAB IV Analisis Kebutuhan

Pada bab ini, membahas tentang analisis kebutuhan sistem informasi penggajian pada PT Mahkota Mitra Sentosa yang terdiri dari kebutuhan fungsional dan kebutuhan non fungsional sistem. Selain itu pada bab ini meliputi pemodelan proses bisnis dan pemodelan sistem.

### BAB V Perancangan Sistem

Bab ini berisi tentang uraian perancangan sistem informasi pada PT Mahkota Mitra Sentosa yang telah digali sebelumnya.

### BAB VI Implementasi

Pada bab ini, akan dibahas mengenai semua proses implementasi sistem informasi sesuai dengan analisis dan perancangan yang telah dibuat.

### BAB VII Pengujian

Dalam bab ini menjelaskan mengenai hasil pengujian sistem informasi yang telah dibuat.

### BAB VIII Penutup

Dalam bab ini menjelaskan tentang kesimpulan dan saran dari proses pengujian yang telah dilakukan serta tindakan yang dapat diambil dari hasil penelitian yang telah dilakukan.

## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Kajian Pustaka

Terdapat dapat beberapa rujukan yang digunakan peneliti untuk membantu dalam menyelesaikan penelitian ini. Pertama, penelitian yang dilakukan oleh Ardhani Dwi Kinasih, dkk (2018) berjudul “Pengembangan Sistem Evaluasi Pembelajaran PAUD (Studi Kasus di PAUD Seruni 05 Kota Malang)”. Dalam penelitian yang dilakukan, peneliti menggunakan pendekatan Rational Unified Process (RUP) sebagai metode pengembangan perangkat lunak dan menggunakan Unified Modeling Language (UML) untuk memodelkan perancangan perangkat lunak yang akan dibangun. Dalam pengembangan sistemnya, peneliti membagi proses pengembangan menjadi empat tahapan utama yaitu tahapan *inception*, *elaboration*, *construction*, dan *transition*. Pada fase *inception* berfokus pada rekayasa kebutuhan fungsi penilaian, kompilasi dan manajemen *user*. Fase *elaboration* berfokus pada penambahan fungsi manajemen murid dan cetak penilaian serta aktivitas rekayasa kebutuhan, perancangan, implementasi, dan pengujian. Fase *construction* dilakukan rekayasa kebutuhan, perancangan, dan implementasi yang berfokus manajemen agenda, indikator, dan tema. Dalam fase *construction* juga dilakukan pengujian integrasi gabungan fungsi *login*, penilaian, dan COMPLIANCE. Sedangkan pengujian validasi berfokus pada manajemen murid dan *user*. Pada fase terakhir yaitu *transition* dilakukan pengujian validasi yang berfokus pada manajemen agenda, indikator, dan tema. Serta pengujian *usability* dan *compatibility* terhadap keseluruhan sistem. Jurnal ini digunakan oleh peneliti untuk membantu dalam pengimplementasian metode *Rational Unified Process* (RUP) sebagai pendekatan dalam pengembangan sebuah sistem informasi dan *Unified Modeling Language* (UML) untuk memodelkan perancangan perangkat lunak yang akan dibangun.

Kedua, penelitian yang dilakukan oleh Sri Rahayu, dkk (2017) berjudul “Perancangan Aplikasi Penggajian Berbasis Web di Sekolah Menengah Kejuruan Islam Atturmudziyyah Garut” . Dalam penelitian tersebut dipaparkan bahwa sistem penggajian yang bersifat manual dengan menggunakan sistem pembukuan sering menimbulkan masalah antara lain hasil *output* yang kurang teliti dalam perhitungan jumlah gaji karyawan serta memakan waktu yang cukup lama dalam prosesnya. Sehingga penelitian ini menghasilkan sebuah sistem informasi penggajian yang bertujuan semua data tersimpan dalam basis data yang akan memudahkan pengolahan data, pengambilan, penyimpanan, pencetakan laporan data guru/pegawai serta efisiensi waktu dalam menyelesaikan pekerjaan. Pengembangan sistem informasi tersebut menggunakan metode pengembangan perangkat lunak *Rational Unified Process* (RUP), namun yang digunakan pada penelitian ini dibatasi hanya sampai tahap *construction*. Jurnal penelitian ini oleh peneliti digunakan membantu dalam pengimplementasian metode *Rational Unified Process* (RUP) sebagai pendekatan dalam pengembangan perangkat lunak serta untuk memperkuat penelitian bahwa dengan mengimplementasikan sistem penggajian akan dapat memudahkan proses penggajian karyawan.

Ketiga, penelitian yang dilakukan oleh Dwi Jayanti, dkk (2014) berjudul "Sistem Informasi Penggajian Pada CV. Blumbang Sejati Pacitan". Dalam penelitian tersebut membahas mengenai sistem pengolahan data penggajian karyawan pada CV. Blumbang Sejati masih bersifat sederhana. Data karyawan tiap bulan masih dicatat dalam sebuah buku khusus, dan untuk menyerahkan bukti laporan gaji masih bersifat konvensional. Masalah yang timbul adalah ketidakkonsistensian data yang dapat menyebabkan kerangkapan data, apabila dalam pencarian data membutuhkan waktu yang sangat lama karena masih menggunakan sistem yang konvensional, dan keamanan data pun masih kurang terjamin. Pihak administrasi pun juga kesulitan dalam proses pembuatan laporan gaji karyawan perbulan untuk diberikan pada Direktur CV. Blumbang Sejati. Berdasarkan masalah tersebut penelitian ini membangun sebuah sistem informasi penggajian menggunakan metode pengembangan *Waterfall*. Sistem informasi penggajian yang dibangun memiliki manfaat yaitu mempercepat dalam proses pembuatan laporan penggajian perbulan dari sistem manual ke sistem komputerisasi yang akan mempermudah pekerjaan dan mengolah penggajian karyawan secara cepat dan tepat waktu. Dengan begitu jurnal penelitian ini digunakan oleh peneliti sebagai rujukan bagaimana mengimplementasikan sistem informasi penggajian yang sesuai dengan kebutuhan dan penguat dasar pentingnya dilakukan penelitian terhadap pengembangan sistem informasi penggajian.

Keempat, penelitian yang dilakukan oleh Lan Jin dan Xiujuan Liang (2012) berjudul "*System Modeling of Vehicle Management Based on RUP and UML*" membahas mengenai pemodelan sistem manajemen kendaraan. Penelitian ini dalam memodelan sistem tersebut menggunakan pendekatan pengembangan perangkat lunak *Rational Unified Process* (RUP) dan *Unified Modeling Language* (UML). Penelitian tersebut memaparkan bahwa dengan menggunakan pendekatan *Rational Unified Process* (RUP) dan *Unified Modeling Language* (UML) dapat membantu pengembang untuk memahami kebutuhan sistem yang akan dibangun serta dapat memandu pengembang untuk dapat mengurangi resiko. Dari hasil penelitian tersebut, jurnal ini digunakan sebagai rujukan supaya membantu peneliti dalam menerapkan metode pengembangan *Rational Unified Process* (RUP) dan *Unified Modeling Language* (UML) serta untuk memperkuat dasar pentingnya menggunakan metode tersebut.

Tabel 2.1 Kajian Pustaka

No.	Nama penulis, jurnal, tahun jurnal, penerbit jurnal	Tujuan penelitian	Metode penelitian / pengembangan	Kesimpulan
1.	<p><b>Nama Penulis:</b> Ardhani Dwi Kinasih, Faizatul Amalia, Bayu Priyambadha</p> <p><b>Nama Jurnal:</b> Pengembangan Sistem Evaluasi Pembelajaran PAUD (Studi Kasus di PAUD Seruni 05 Kota Malang)</p> <p><b>Tahun jurnal:</b> 2018</p> <p><b>Penerbit jurnal:</b> Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer</p>	<p>Penelitian ini bertujuan merancang dan mengimplementasikan metode RUP dalam pengembangan Sistem Evaluasi Pembelajaran di PAUD Seruni 05. Serta melakukan analisis terhadap hasil pengujian untuk mengetahui kualitas sistem.</p>	<p>Penelitian menerapkan tipe penelitian implementatif pengembangan atau <i>research and development (R &amp; D)</i>. perangkat lunak yang dikembangkan menggunakan <i>web platform</i> dan PHP. Alur penelitiannya di mulai dengan studi kepustakaan terkait penilaian PAUD, MYSQL, PHP, pemodelan UML, serta implementasi metode pengembangan perangkat lunak <i>Rational Unified Process (RUP)</i> yang terdiri dari empat fase yaitu <i>inception, elaboration, construction</i>, dan <i>transition</i>.</p>	<p>Perancangan Sistem Evaluasi Pembelajaran PAUD menerapkan pemodelan UML dan perancangan sistem menyesuaikan dengan rekayasa kebutuhan sistem fungsional dan kebutuhan non fungsional. Dengan pengembangan sistem menerapkan metode pengembangan perangkat lunak RUP. Hasil dari pengujian unit dan pengujian integrasi menunjukkan bahwa logika program mudah dipahami dan mudah untuk di maintenance. Pengujian validasi menunjukkan seluruh fungsi pada sistem telah valid. Pengujian usability menunjukkan tingkat kemudahan pengguna untuk menggunakan sistem. Pengujian <i>compatibility</i> menunjukkan bahwa sistem dapat berjalan disemua perangkat android minimal versi 2.3</p>
2.	<p><b>Nama penulis:</b> Sri Rahayu, Mochammad Fajar Hadian Subagja</p>	<p>Tujuan dari penelitian yang dilakukan yaitu untuk membuat aplikasi</p>	<p>Metode perancangan yang digunakan yaitu <i>Rational Unified Process (RUP)</i> yang</p>	<p>Hasil dari penelitian ini berupa aplikasi penggajian di Sekolah Menengah Kejuruan Islam Atturmudziyyah Garut</p>

No.	Nama penulis, jurnal, tahun jurnal, penerbit jurnal	Tujuan penelitian	Metode penelitian / pengembangan	Kesimpulan
	<p><b>Nama jurnal:</b> Perancangan Aplikasi Penggajian Berbasis Web Di Sekolah Menengah Kejuruan Islam Atturmudziyyah Garut</p> <p><b>Tahun jurnal:</b> 2017</p> <p><b>Penerbit jurnal:</b> Jurnal Algoritme Sekolah Tinggi Teknologi Garut</p>	<p>penggajian di Sekolah Menengah Kejuruan Islam Atturmudziyyah Garut sehingga dapat membantu mempercepat dan memudahkan pengelolaan gaji guru atau pegawai khususnya menginformasikan data laporan gaji bulanan kepada Kepala Sekolah.</p>	<p>merupakan metodologi pengembangan sistem berbasis objek dengan tahapan meliputi: <i>inception, elaboration, construction</i>, dan <i>transition</i>. Dalam pemodelan perancangan digunakan <i>Unified Modeling Language (UML)</i> dengan diagram yang digunakan meliputi <i>use-case diagram, activity diagram, sequence diagram</i>, dan <i>class diagram</i>. Pengujian yang dilakukan yaitu menggunakan <i>black-box testing</i>.</p>	<p>ini memiliki fasilitas untuk pengolahan data gaji guru/pegawai dilakukan secara otomatis. Semua data tersebut tersimpan dalam basis data yang akan memudahkan pengolahan data, pengambilan, penyimpanan, mencetak laporan data guru/pegawai, mencetak slip gaji jadi tidak lagi mengantri sehingga efisiensi waktu dalam menyelesaikan pekerjaan.</p>
3.	<p><b>Nama penulis:</b> Dwi Jayanti, Siska Iriani</p> <p><b>Nama jurnal:</b> Sistem Informasi Penggajian Pada CV. Blumbang Sejati Pacitan</p> <p><b>Tahun jurnal:</b> 2014</p>	<p>Tujuan dari penelitian ini yaitu menghasilkan program aplikasi untuk mengolah data penggajian pada CV. BLUMBANG SEJATI</p>	<p>Metode yang digunakan dalam penelitian ini yaitu metode interview, studi literatur, analisis sistem, perancangan sistem, implementasi, verifikasi, dan maintenance</p>	<p>Hasil dari penelitian ini berupa sistem informasi penggajian yang membantu dalam proses pengelolaan data gaji karyawan, sehingga admin memperoleh kemudahan dalam melakukan operasi sistem karena sistem yang telah dibuat sangat mudah digunakan. Sistem penggajian karyawan yang terkomputerisasi dapat mempermudah dalam pencarian data</p>



No.	Nama penulis, jurnal, tahun jurnal, penerbit jurnal	Tujuan penelitian	Metode penelitian / pengembangan	Kesimpulan
	<b>Penerbit jurnal:</b> Jurnal Speed - Sentra Penelitian Engineering dan Edukasi			daripada menggunakan sistem yang konvensional. Penggunaan sistem ini membantu pengolahan data gaji pada CV. Blumbang Sejati Pacitan menjadi lebih efektif dan efisien.
4.	<p><b>Nama penulis:</b> Lan Jin, Xiujuan Liang</p> <p><b>Nama jurnal:</b> System Modeling of Vehicle Management Based on RUP and UML</p> <p><b>Tahun jurnal:</b> 2012</p> <p><b>Penerbit jurnal:</b> IEEE</p>	Tujuan penelitian ini yaitu menganalisis sistem, desain, dan membangun model untuk sistem manajemen kendaraan.	Metode pengembangan yang digunakan dalam penelitian ini yaitu metode pengembangan <i>Rational Unified Process</i> (RUP) menggunakan 3 <i>workflow</i> yaitu <i>pemodelan bisnis, analisis persyaratan, dan desain</i> . Serta UML dalam menggambarkan diagram <i>use-case diagram, class diagram, dan Sequence diagram</i> .	Hasil dari penelitian ini menunjukkan bahwa RUP menyediakan kriteria yang diperlukan dan panduan alat dalam aspek sebagai <i>use-case</i> driven, pengembangan iterasi, pemodelan visual. Penggunaan UML dapat membantu pengembang untuk memiliki pemahaman yang jelas mengenai keseluruhan sistem. Menggabungkan RUP dengan UML memandu pengembang perangkat lunak dapat mengurangi resiko, menstandarisasi proses manajemen dan pengembangan, dan meningkatkan kualitas, efisiensi, dan pemeliharaan pengembang perangkat lunak

## 2.2 PT Mahkota Mitra Sentosa

PT Mahkota Mitra Sentosa merupakan salah satu perusahaan yang bergerak di bidang jasa pelayanan transportasi guna memenuhi kebutuhan transportasi di era sekarang ini, khususnya di dunia pariwisata. PT Mahkota Mitra berlokasi di Jln. R.A. Kartini 121 Kayenlor Plemahan Kabupaten Kediri. Saat ini PT Mahkota Mitra mempunyai armada sekitar 25. PT Mahkota Mitra Sentosa mempunyai visi yaitu akan menjadi salah satu perusahaan yang mampu menjawab dan memenuhi kebutuhan masyarakat di dalam melaksanakan perjalanan wisata. Misinya yaitu akan bekerja secara professional, mengutamakan keselamatan, kenyamanan, dan sangat memperhatikan kepuasan konsumen.

## 2.3 Proses Bisnis

Menurut Monk & Wagner (2013), proses bisnis adalah serangkaian aktivitas di dalamnya terdapat *input* dan menghasilkan sebuah *output* yang bernilai bagi *customer*. Dalam hal ini, pelanggan dibedakan menjadi pelanggan eksternal dan internal. Pelanggan eksternal merupakan seseorang atau kelompok yang membeli produk. Sedangkan pelanggan internal merupakan seseorang atau kelompok yang berada pada departemen dalam perusahaan. Memahami aktivitas sebuah organisasi atau perusahaan dapat menggunakan pemodelan proses bisnis. Terdapat beberapa pendekatan untuk mengetahui proses bisnis yaitu dengan melakukan proses analisis dan memodelkan proses bisnis *as-is* dan proses bisnis *to-be*.

### 2.3.1 Analisis dan Pemodelan Proses Bisnis As-Is

Analisis dan pemodelan proses bisnis *as-is* merupakan aktivitas mengumpulkan dan memodelkan proses bisnis yang sedang berjalan di perusahaan dengan tujuan untuk menyajikan informasi mengenai situasi aktual dan mengevaluasi kekurangan sehingga dapat merekomendasikan perbaikan proses bisnis yang ada pada perusahaan. Identifikasi proses bisnis *as-is* dapat dilakukan dengan menggunakan metode wawancara kepada pihak terkait dari proses bisnis. Selain itu, dapat diidentifikasi menggunakan metode observasi terhadap pihak terkait yang sedang melakukan tugasnya (Becker, et al., 2013). Hasil dari indentifikasi tersebut selanjutnya digunakan sebagai sketsa diagram. Sketsa diagram digunakan untuk memodelkan proses bisnis *as-is*. Kegiatan yang kompleks dapat dimodelkan menggunakan sub-proses. Setelah dilakukan analisis dan pemodelan untuk memastikan bahwa proses bisnis yang dimodelkan telah sesuai maka dibutuhkan persetujuan pemangku kepentingan (Przybyłek, 2011).

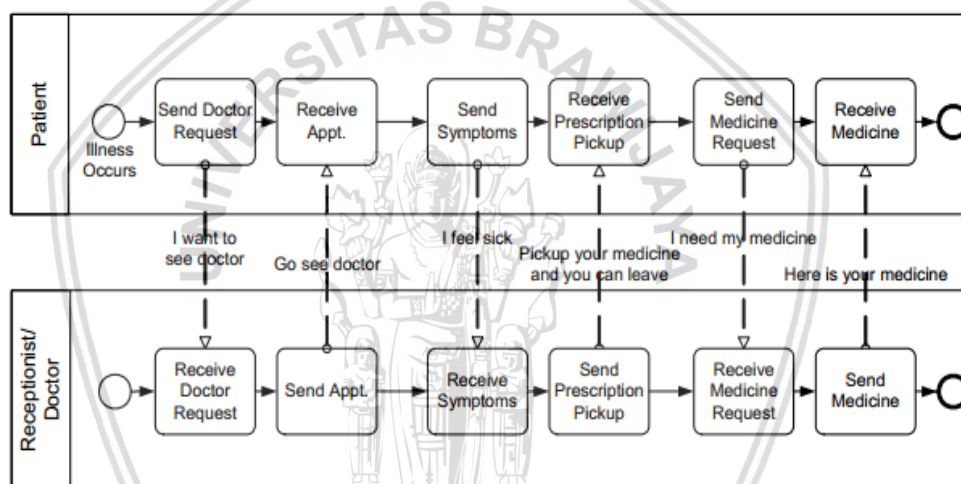
### 2.3.2 Analisis dan Pemodelan Proses Bisnis To-Be

Analisis dan pemodelan proses bisnis *to-be* merupakan aktivitas mengidentifikasi dan memodelkan peningkatan proses bisnis berdasarkan analisis proses bisnis *as-is*. Perubahan dari proses bisnis *as-is* menjadi proses bisnis *to-be* dapat dilakukan secara bertahap. Proses pemodelan proses bisnis *to-be* melibatkan pemangku kepentingan dengan tujuan supaya proses bisnis yang

dikembangkan tetap sesuai kebutuhan pemangku kepentingan (Becker, et al., 2013). Penerapan otomatisasi pada proses bisnis *to-be* mungkin dilakukan. Namun, penerapan tersebut tidak sepenuhnya menjamin terjadinya peningkatan yang signifikan. Oleh karena itu penting melibatkan pemangku kepentingan dalam pemodelan proses bisnis *to-be* (Przybyłek, 2011).

## 2.4 Bussiness Process Model dan Notation

Menurut *Object Management Group* (2011), BPMN atau *Business Process Model and Notation* digunakan sebagai jembatan standarisasi untuk memenuhi kesenjangan antara desain proses bisnis dan pelaksanaan proses bisnis. Pengembangan BPMN bertujuan untuk menyediakan notasi yang mudah dibaca oleh seluruh pelaku bisnis, dari analisis bisnis yang membuat daftar proses hingga pengembangan teknis yang bertanggung jawab dalam mengimplementasi teknologi yang akan menjalankan proses bisnis dan bagi para pelaku bisnis yang akan menjalankan dan mengelola proses tersebut.







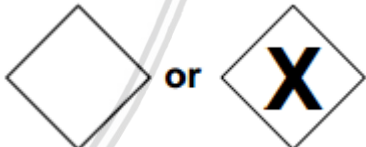

Gambar 2.1 Contoh BPMN pada proses bisnis aplikasi dokter






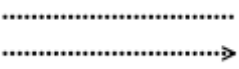
Sumber: (BPMN, 2011)

Element-element yang terdapat dalam BPMN, yaitu:





Tabel 2.1 Elemen pada BPMN (Object Mangement Group, Inc. 2011)

Elemen	Notasi	Deskripsi
<i>Event</i>	Mendefinisikan sesuatu yang terjadi selama proses berjalan.	
<i>Flow Dimension (Start)</i>	Mendefinisikan event yang menandakan proses dimulai.	

Elemen	Notasi	Deskripsi
<i>Flow Dimension (Intermediate)</i>	Mendefinisikan event yang berada antara start dan end. Notasi ini akan berdampak pada proses, tetapi tidak memulai proses ( secara langsung ) atau mematikan proses tersebut.	
<i>Flow Dimension (End)</i>	Mendefinisikan event yang menandakan proses dalam BPMN berakhir	
<i>Activity</i>	Mendefinisikan istilah generik untuk pekerjaan yang dilakukan organisasi dalam sebuah proses	
<i>Gateway</i>	Mendefinisikan terkait suatu percabangan dan penggabungan jalur. Penanda internal akan menunjukkan jenis kontrol perilaku	
<i>Gateway Control Type (Exclusive)</i>	Mendefinisikan mengenai jalur alternatif BPMN. Tetapi, terdapat satu jalur atau arah yang dapat dilalui.	
<i>Gateway Control Type (Inclusive)</i>	Mendefinisikan mengenai jalur alternatif BPMN. Keseluruhan jalur yang ada dapat atau bahkan tidak satupun jalur dapat dilalui. Namun, tetap harus didesain sedemikian rupa sehingga paling tidak terdapat satu	

Elemen	Notasi	Deskripsi
	jalur pasti akan dilalui dalam proses bisnis.	
<i>Gateway Control Type (Event Based)</i>	Mendefinisikan beberapa jalur yang tersedia ketika ada suatu event atau kejadian. Notasi ini hanya terdapat satu jalur yang dapat dilalui. Namun, yang dicek bukan merupakan kondisi melainkan event apa yang terjadi akan menentukan jalur yang diambil proses bisnis.	
<i>Gateway Control Type (Parallel Event-Based)</i>	Notasi ini sama seperti event based. Namun, memungkinkan jalur yang ada dalam proses dilalui secara bersamaan.	
<i>Gateway Control Type (Parallel)</i>	Mendefinisikan terdapat beberapa jalur dalam proses bisnis. Keseluruhan jalur dilalui secara bersamaan.	
<i>Sequence Flow</i>	Mendefinisikan urutan aktivitas yang akan dijalankan.	
<i>Message Flow</i>	Mendefinisikan pesan atau informasi antar dua partisipan yang saling bertukar pesan.	
<i>Association</i>	Digunakan untuk menghubungkan antara informasi dan artefak dengan elemen grafis BPMN.	



Elemen	Notasi	Deskripsi
<i>Pool</i>	Merepresentasikan dari partisipan dalam BPMN.	
<i>Lane</i>	Merupakan partisi dalam proses, dalam pool dan akan menambah proses (horizontal atau vertikal).	
<i>Data Object</i>	Mendefinisikan informasi mengenai aktivitas yang dibutuhkan untuk dijalankan dan apa yang dihasilkan oleh aktivitas	
<i>Message</i>	Mendefinisikan sebagai gambaran komunikasi antar partisipan.	

## 2.5 Sistem

Menurut Abdul Kadir (2013) dalam bukunya yang berjudul pengenalan sistem informasi, sistem merupakan serangkaian elemen yang saling terkait untuk mencapai suatu tujuan. Dengan gambaran, jika sebuah sistem terdapat elemen yang tidak memberikan manfaat untuk mencapai tujuan sama, maka elemen tersebut dipastikan bukanlah merupakan bagian dari sistem.

Sedangkan menurut O'Brien dan Marakas (2010), sistem adalah seperangkat komponen yang saling terkait, memiliki batas jelas, untuk mencapai tujuan yang telah ditetapkan dengan mengolah input dan menghasilkan output yang terorganisir. Sistem memiliki tiga fungsi utama antara lain sebagai berikut:

1. *Input* : Meliputi pengambilan dan perakitan elemen yang masuk ke dalam sistem untuk kemudian diproses.
2. *Processing* : Meliputi proses transformasi untuk mengubah input menjadi output.
3. *Output* : Pemindahan elemen yang telah diolah dari proses transformasi ke dalam bentuk yang bisa digunakan dan sesuai dengan tujuan akhir pengolahan elemen.

Dari beberapa definisi di atas dapat disimpulkan bahwa sistem adalah kumpulan komponen-komponen yang saling berhubungan untuk mencapai suatu tujuan.

## 2.6 Informasi

Menurut McFadden, dkk. (1999) dalam Abdul Kadir (2013) mendefinisikan informasi sebagai data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut. Sedangkan menurut Davis (1999) dalam Abdul Kadir (2013), informasi merupakan data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermaksud dalam pengambilan keputusan saat ini dan masa mendatang.

Sehingga dalam garis besar informasi dapat diartikan sebagai adalah kumpulan fakta – fakta yang telah diolah sehingga memiliki nilai dan bermanfaat bagi penggunaannya.

## 2.7 Sistem Informasi

Aktivitas orang dan teknologi informasi yang dikombinasikan menggunakan teknologi tersebut untuk mendukung operasi dan manajemen. Istilah sistem informasi merujuk kepada interaksi antara orang, data, proses algoritme, dan teknologi. Serta istilah ini tidak hanya merujuk penggunaan organisasi teknologi informasi dan komunikasi, tetapi cara di mana seseorang atau kelompok berinteraksi menggunakan teknologi dalam mendukung proses bisnis (Kroenke, 2008).

Sistem Informasi merupakan kombinasi dari manusia, alat teknologi, prosedur, media, dan pengendalian yang bertujuan menata jaringan komunikasi yang penting. Proses dari transaksi tertentu dan rutin, dapat membantu manajemen dalam melakukan pengambilan keputusan yang tepat (Rizki, et al., 2014).

Sistem informasi adalah kombinasi dari *software*, *hardware*, dan jaringan telekomunikasi yang dibangun dan digunakan orang untuk mengumpulkan, membuat, dan mendistribusikan data yang bermanfaat. Namun, cara terbaik untuk menggunakan sistem informasi adalah mendukung strategi organisasi dengan cara yang memungkinkan perusahaan untuk memperoleh atau mempertahankan keunggulan kompetitif dibandingkan pesaingnya (Valacich & Schneider, 2012).

Sehingga dapat disimpulkan sistem informasi adalah sistem yang dapat mengelola suatu data baik yang mempunyai kapasitas sederhana maupun yang kompleks untuk diubah menjadi sebuah informasi yang digunakan untuk membantu dalam pengambilan keputusan.

## 2.8 Sistem Penggajian

Sistem penggajian dibagi menjadi tiga sistem yaitu (Hasibuan, 2010):

1. Sistem waktu

Dalam ini besarnya gaji didapatkan berdasarkan standar waktu seperti jam, minggu, atau bulan. Administrasi pengubahan sistem waktu relatif mudah serta dapat diterapkan kepada karawan tetap maupun pekerja harian. Besarnya gaji hanya didasarkan kepada lamanya bekerja. Kelebihan menggunakan sistem ini adalah administrasi pengupahan mudah serta besarnya gaji yang dibayarkan tetap. Kekurangan sistem waktu adalah pekerja yang malas pun besarnya gaji tetap dibayar sebesar perjanjian.

2. Sistem hasil (*output*)

Dalam ini, besarnya gaji yang dibayar berdasarkan banyaknya hasil uang didapatkan. Sistem hasil ini tidak dapat diterapkan bagi karyawan tetap dan jenis pekerjaan yang tidak mempunyai standar fisik, seperti bagi karyawan administrasi. Kelebihan menggunakan sistem ini memberikan kesempatan bagi karyawan yang bekerja dengan sungguh-sungguh serta berprestasi baik maka mendapatkan balas jasa yang lebih besar. Kekurangan dari sistem ini yaitu kualitas barang yang dihasilkan terkadang kurang baik serta karyawan yang kurang mampu mendapatkan balas jasanya kecil, sehingga dalam menggunakan sistem ini dianggap kurang manusiawi untuk karyawan.

3. Sistem borongan

Sistem ini merupakan suatu sistem yang cara pengupahannya penetapan besarnya jasa didasarkan dari volume dan lama pengerjaan yang telah dilakukan. Penetapan besarnya balas jasa cukup rumit, lama dalam mengerjakannya, serta banyak alat yang digunakan untuk menyelesaikannya.

## 2.9 Vision Document

*Vision document* merupakan artefak dari *Rational Unified Process* berupa dokumen awal untuk menangkap pemahaman yang berkaitan dengan pemangku kepentingan, permasalahan, dan kebutuhan saat ini. *Vision document* merupakan media komunikasi yang digunakan antara tim manajemen dengan tim proyek dalam pembangunan sebuah sistem informasi (Bittner & Spence, 2002).

### 2.9.1 Pemangku Kepentingan

Pemangku kepentingan merupakan seseorang yang terpengaruh oleh hasil keluaran dari sistem atau proyek yang menghasilkan sistem (Bittner & Spence, 2002). Berdasarkan definisi tersebut, pada lingkup pengembangan perangkat lunak didapatkan dua pemangku kepentingan, antara lain pengguna sistem dan tim pengembang.

### 2.9.2 Tipe Pemangku Kepentingan

Melakukan klasifikasi tipe pemangku kepentingan merupakan langkah yang harus dilakukan untuk memahami pemangku kepentingan. Klasifikasi

dilakukan dengan mengelompokkan pemangku kepentingan yang memiliki karakteristik dan hubungan yang sama. Berikut tipe – tipe pemangku kepentingan, antara lain (Bittner & Spence, 2002):

- Pengguna : Merupakan pengguna langsung sistem. Perannya akan didefinisikan secara langsung oleh aktor sistem.
- Sponsor : Merupakan pemangku kepentingan yang memberikan investasi terhadap jalannya proyek.
- Pengembang: Merupakan pemangku kepentingan yang bertugas untuk menjalankan proyek.
- Pemegang otoritas : Merupakan ahli dalam aspek khusus pada suatu domain yang biasanya berkaitan dengan pemerintah, pemberi wewenang dan aturan, serta bagian legislatif pada suatu lembaga.
- Pelanggan : Merupakan pemangku kepentingan yang akan membeli sistem.

### 2.9.3 Perwakilan dan Peran Pemangku Kepentingan

Langkah selanjutnya yaitu mendefinisikan peran yang akan dilakukan oleh pemangku kepentingan dalam proses pengembangan proyek. Sebelum peran pemangku kepentingan diidentifikasi, dilakukan aktivitas penentuan perwakilan pemangku kepentingan. Perwakilan pemangku kepentingan merupakan anggota komunitas dari pemangku kepentingan yang terlibat langsung pada proses pengembangan proyek. Peran bagi pemangku kepentingan yaitu klasifikasi dari tanggung jawab pemangku kepentingan kepada proyek yang dikerjakan. Hal ini diperlukan supaya pemangku kepentingan mengetahui tingkat keterlibatan dalam pengembangan proyek (Bittner & Spence, 2002).

### 2.9.4 Pernyataan Permasalahan

Permasalahan merupakan adanya perbedaan antara hal yang didapatkan dengan hal yang diharapkan (Bittner & Spence, 2002). Untuk memenuhi kebutuhan pengguna perlu memahami apa permasalahan mereka, supaya kedepannya mendapatkan solusi yang dapat menyelesaikan permasalahan pengguna. Menangkap permasalahan yang ada dilakukan dengan cara membuat pernyataan permasalahan. Hal tersebut merupakan ringkasan dari pemahaman permasalahan yang dialami oleh pengguna, akibat dari permasalahan, serta keuntungan yang didapatkan dari solusi penyelesaiannya.

Berikut *template* yang digunakan untuk menangkap permasalahan (Bittner & Spence, 2002).

**Tabel 2.2 Pernyataan Masalah yang Telah Diadaptasi (Bittner & Spence, 2002)**

Permasalahan	[Deskripsi Permasalahan]
--------------	--------------------------

**Tabel 2.2 Pernyataan Masalah yang Telah Diadaptasi (Bittner & Spence, 2002)**

Memengaruhi	[pemangku kepentingan yang terpengaruh oleh permasalahan]
Akibat dari permasalahan	[apa akibat dari permasalahan?]
Solusi diharapkan dapat	[daftar keuntungan dari solusi yang berhasil]

Pada Tabel 2.2 merupakan tabel pernyataan masalah yang digunakan untuk menangkap permasalahan yang dihadapi oleh pemangku kepentingan beserta akibat serta keuntungan dari solusi yang diterapkan. Sehingga dengan mengetahui hal tersebut, dapat diketahui asal mula dari kebutuhan pemangku kepentingan (Bittner & Spence, 2002).

### 2.9.5 Kebutuhan Pemangku Kepentingan

Kebutuhan pemangku kepentingan merupakan gambaran dari permasalahan atau kesempatan pada bisnis, personal yang ditujukan untuk memberi alasan mengenai pertimbangan, pembelian, atau penggunaan sistem baru. Penjelasan mengenai kebutuhan pemangku kepentingan memudahkan analisis untuk memahami permasalahan yang berbeda dapat memberi efek yang berbeda kepada pemangku kepentingan. Kebutuhan pemangku kepentingan memberikan pemahaman atas permasalahan yang lebih spesifik daripada pernyataan masalah (Bittner & Spence, 2002). Pada pendefinisian kebutuhan pemangku kepentingan, disarankan untuk menggunakan pengkategorian aturan dari MoSCoW. Aturan MoSCoW digunakan untuk mendefinisikan prioritas dari kebutuhan yang didapatkan. Berikut ini merupakan aturan MoSCoW mengenai prioritas antara lain (DSDM, 2014):

- *Mo (Must Have)* : Dapat diartikan jika kebutuhan pengguna tidak dapat diselesaikan oleh sistem maka sistem yang dibangun tidak dapat menjadi solusi bagi kebutuhan pengguna.
- *S (Should Have)* : Dapat diartikan jika kebutuhan pengguna tidak diselesaikan oleh sistem maka sistem yang dibangun masih bisa menjadi solusi dari kebutuhan pengguna, namun hal tersebut akan memberatkan bagi pengguna.
- *Co (Could Have)* : Dapat diartikan jika kebutuhan pengguna tidak diselesaikan oleh sistem maka sistem yang dibangun masih bisa menjadi solusi dari kebutuhan pengguna. Hal tersebut tidak akan memberatkan bagi pengguna.
- *W (Won't Have)* : Dapat diartikan kebutuhan pengguna yang dapat didefinisikan sebagai solusi dari permasalahan, namun kebutuhan tersebut



disepakati oleh tim pengembang untuk tidak digunakan pada sistem yang dibangun.

### 2.9.6 Kedudukan Produk

Sistem membutuhkan sebuah pernyataan yang baik atau alasan mengapa sistem tersebut dibangun. Oleh karena itu perlu dinyatakan secara jelas apa yang dikerjakan oleh sistem dan mengapa hal itu harus dikerjakan oleh sistem. Untuk mendefinisikan pernyataan tersebut, dapat menggunakan kedudukan produk. Kedudukan produk telah menyediakan informasi tentang asumsi, kemampuan, dan alternatif pada suatu produk (Bittner & Spence, 2002). Tabel 2.3 merupakan pernyataan kedudukan produk yang dapat mendefinisikan esensi dari apa yang dikerjakan oleh sistem.

**Tabel 2.2 *Template* Pernyataan Kedudukan Produk yang telah Diadaptasi (Bittner & Spence, 2002)**

Untuk	[pengguna target]
Yang	[kebutuhan pengguna]
Produk	[nama produk]
Yang	[keuntungan produk]
Tidak Seperti	[alternatif lain]
Produk Kami	[perbedaan dengan alternatif]

### 2.9.7 Fitur

Fitur merupakan kemampuan yang dimiliki oleh sistem yang diperlukan untuk memenuhi kebutuhan pengguna. Setiap fitur yang didefinisikan selanjutnya akan menjadi dasar dalam pendefinisian *use-case* (Bittner & Spence, 2002). Untuk mendokumentasikan fitur, dapat menggunakan *template* yang dibuat oleh (Bittner & Spence, 2002).

**Tabel 2.3 *Template* Fitur yang telah Diadaptasi (Bittner & Spence, 2002)**

Kode	Deskripsi	Prioritas
FEAT1	Deskripsi fitur	Mo/S/Co/W

Tabel 2.4 merupakan *template* fitur yang diadaptasi dari Bittner & Spence (2002). Dalam penulisan fitur dianjurkan menggunakan kode unik, mencantumkan deskripsi setiap fitur, serta memberikan kategori prioritas sesuai kebutuhan yang ada menggunakan aturan dari MoSCoW.

## 2.10 Basis Data dan DBMS

Basis data atau biasa disebut sebagai *database* merupakan suatu pengorganisasian dari sekumpulan data yang saling terkait untuk memudahkan aktivitas dalam memperoleh informasi. Basis data dimaksudkan untuk mengatasi *problem pada* sistem dengan memakai pendekatan berbasis berkas. Pengelolaan basis data diperlukan perangkat lunak yang disebut *Database Management System* (DBMS). DBMS adalah perangkat lunak sistem yang memungkinkan para pengguna dapat membuat, memelihara, mengontrol, dan mengakses basis data dengan cara yang praktis dan efisien. DBMS dapat digunakan untuk mengelola berbagai macam pengguna yang memiliki kebutuhan akses yang berbeda-beda (Kadir, 2013).

## 2.11 Unified Modeling Language (UML)

*Unified Modeling Language* (UML) merupakan ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’. Pemodelan ini digunakan sebagai penyederhanaan permasalahan yang kompleks sedemikian rupa sehingga membuat lebih mudah untuk dipelajari dan dipahami (Nugroho, 2010).

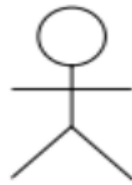
*Unified Modeling Language* merupakan bahasa standar yang digunakan untuk penulisan *blueprint software*. UML digunakan dalam mendefinisikan spesifikasi, mendokumentasikan artifak, memvisualisasikan *construction* dari *software* (Pressman, 2010). Dapat dikatakan, *blueprint* digunakan oleh arsitek dalam perancangan sebuah gedung, maka UML digunakan untuk merancang perangkat lunak dalam pengembangan perangkat lunak.

UML terdiri dari beberapa diagram yang digunakan untuk memvisualisasikan perancangan sistem. Pada penelitian ini menggunakan beberapa diagram UML antara lain:

1. *Use-case Diagram*
2. *Class Diagram*
3. *Activity Diagram*
4. *Sequence diagram*

### 2.11.1 Use-case Diagram

*Use-case diagram* digunakan untuk mengetahui fungsi atau fitur pada perangkat lunak menurut prespektif pengguna (Pressman, 2010). Dalam membuat *use-case diagram*, perlu mengidentifikasi semua hal yang dapat dilakukan oleh pengguna. Kemudian fungsi atau fitur tersebut dikelompokkan berdasarkan jenis penggunaannya. Lalu fungsi atau fitur tersebut dinotasikan dengan notasi *use-case*.



Actor

**Gambar 2.2 Notasi Aktor**

Sumber: (IBM, 2004)

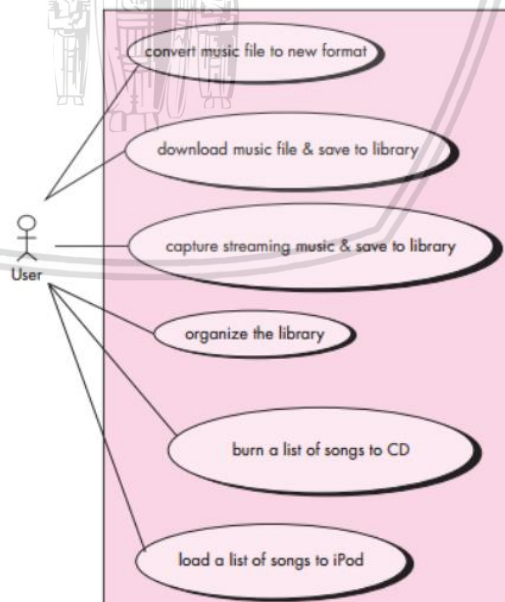
Gambar 2.2 merupakan notasi untuk merepresentasikan aktor sistem. Aktor adalah peran dari pengguna sistem yang di"mainkan" ketika pengguna tersebut menggunakan sistem. Aktor tersebut merepresentasikan manusia, sistem lain, maupun perangkat keras yang berinteraksi dengan sistem tersebut (IBM, 2004).

Use Case

**Gambar 2.3 Notasi Use-Case**

Sumber : (IBM, 2004)

Gambar 2.3 merupakan notasi untuk merepresentasikan *use-case* sistem yang dijalankan oleh aktor. Nama *use-case* yang ditulis menjelaskan apa yang dapat *use-case* tersebut jalankan, bukan bagaimana hal itu dikerjakan (IBM, 2004).

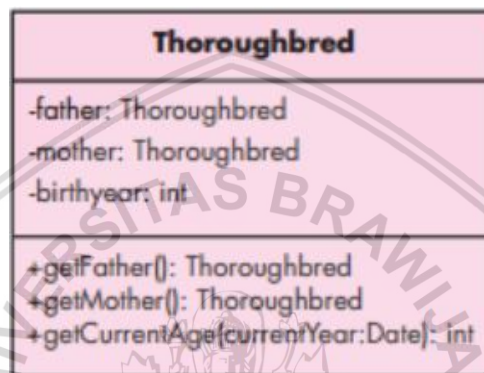
**Gambar 2.4 Use-Case Diagram pada sistem musik**

Sumber: (Pressman, 2010)

Gambar 2.4 dapat diketahui aktor dari perancangan sistem tersebut adalah *user*. Fungsi atau fitur yang dapat dilakukan oleh *user* digambarkan dengan notasi berbentuk oval yang memiliki garis berhubungan dengan aktor *user*. Setiap fungsi atau fitur yang ada dalam *use-case* penulisannya menggunakan kata kerja tanpa subjek.

### 2.11.2 Class Diagram

*Class diagram* dalam perancangan sistem digunakan untuk memodelkan kelas, termasuk atribut, operasi, dan relasi antar kelas lain (Pressman, 2010). Elemen utama dari *class diagram* yaitu segi empat yang dibagi secara horizontal. Berikut contoh gambar dari *class diagram*.



Gambar 2.5 Class Diagram kelas *Thoroughbred*

Sumber: (Pressman, 2010)

Bagian pertama merupakan nama kelas. Pada umumnya nama kelas mewakili sebuah obyek yang ada pada sistem. Kotak kedua adalah atribut yang dimiliki oleh sebuah obyek sistem. Kotak ketiga berisikan operasi yang ada pada kelas tersebut. Operasi merupakan segala macam aktivitas yang dilakukan oleh obyek, dan dapat memiliki tipe data dari hasil keluaran operasi tersebut. Terdapat beberapa notasi pada kelas yaitu -, #, ~, + yang secara berurutan mewakili *private*, *protected*, *package*, dan *public* (Pressman, 2010).



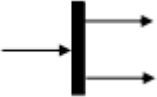


### 2.11.3 Activity Diagram

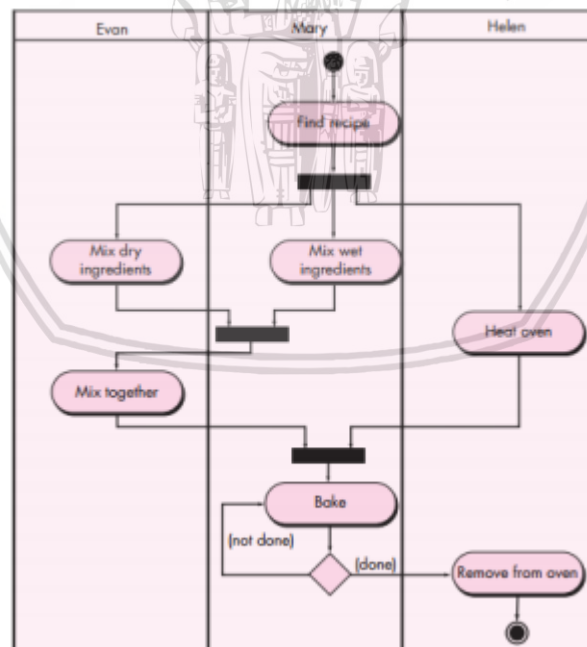
*Activity Diagram* menjelaskan perilaku dinamis dari sistem melalui aliran proses yang dilakukan oleh sistem. Hal ini sama dengan model *flow chart* tetapi sedikit berbeda karena aliran proses dapat ditampilkan secara bersamaan pada *activity diagram* (Pressman, 2010). Berikut adalah simbol-simbol yang digunakan pada *activity diagram*:

Tabel 2.3 Simbol pada *activity diagram*

No	Simbol	Deskripsi
1	●	<i>Initial Node</i> . Menunjukan titik awal kegiatan.

Tabel 2.3 Simbol pada *activity diagram*

No	Simbol	Deskripsi
2		<i>Final Node</i> . Menunjukan akhir kegiatan
3		<i>State/Activities</i> . Aktivitas yang dilakukan oleh sistem maupun <i>user</i> dalam sistem
4		<i>Fork</i> (Percabangan). Mewakili pemisahan aktivitas yang bersamaan menjadi dua atau lebih.
5		<i>Join</i> (Penggabungan) Simbol untuk penggabungan proses
6		<i>Decision</i> . Simbol untuk alur kontrol bercabang yang dikarenakan adanya kondisi.



Gambar 2.6 *Activity Diagram*

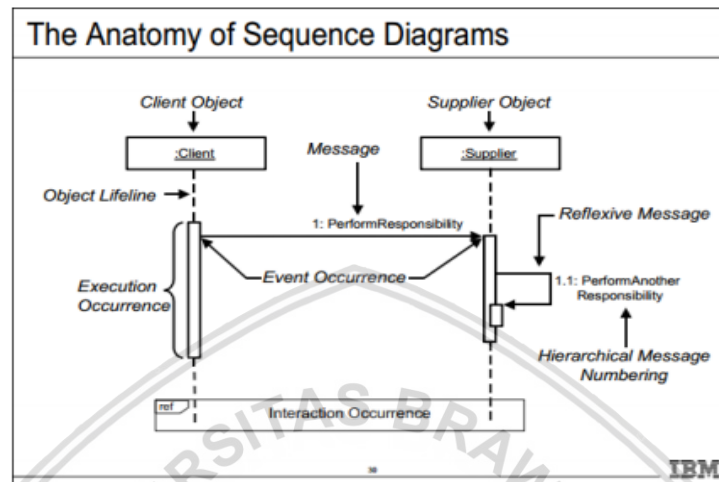
Sumber: (Pressman, 2010)

Gambar lima merupakan contoh dari *activity diagram*. Aktivitas tersebut dimulai dari Mary dan berakhir pada Helen.



#### 2.11.4 Sequence diagram

*Sequence diagram* menunjukkan interaksi antar objek perangkat lunak selama suatu kegiatan berlangsung. Diagram ini menunjukkan urutan fungsi yang dijalankan selama proses *scenario*. Untuk kelas dan fungsi yang dijalankan, didapat dari tahap pendefinisian kelas (Pressman, 2010).

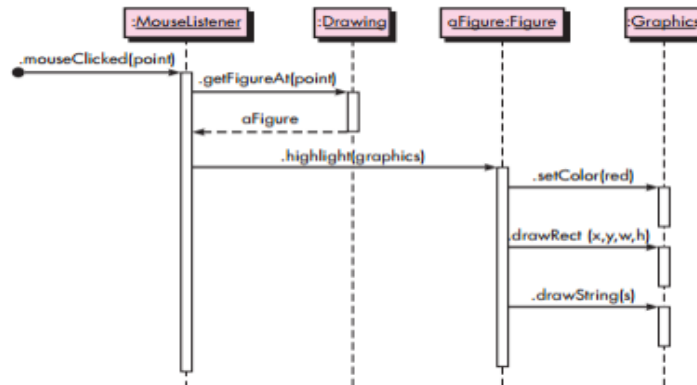


Gambar 2.7 Elemen-elemen pada *Sequence diagram*

Sumber: (IBM, 2004)

Gambar lima merupakan elemen-elemen yang terdapat pada *sequence diagram*. Berikut penjelasan mengenai elemen-elemen tersebut (IBM, 2004):

- *Client Object* : pendefinisian obyek yang melakukan fungsi kepada supplier object.
- *Object Lifeline* : representasi waktu hidup sebuah obyek.
- *Execution Occurrence* : menunjukkan awal dan akhir dari unit perilaku yang dipicu oleh pesan tertentu.
- *Supplier Object* : obyek yang menyediakan suatu fungsi untuk dapat diakses oleh Client Object.
- *Reflexive Message* : pesan yang dikirimkan oleh obyek itu sendiri.
- *Hierarchical Message Numbering* : penomoran pada *sequence* berupa hirarki.
- *Interactiion Occurance* : referensi menuju diagram *sequence* yang lain.



Gambar 2.8 Contoh *Sequence diagram*

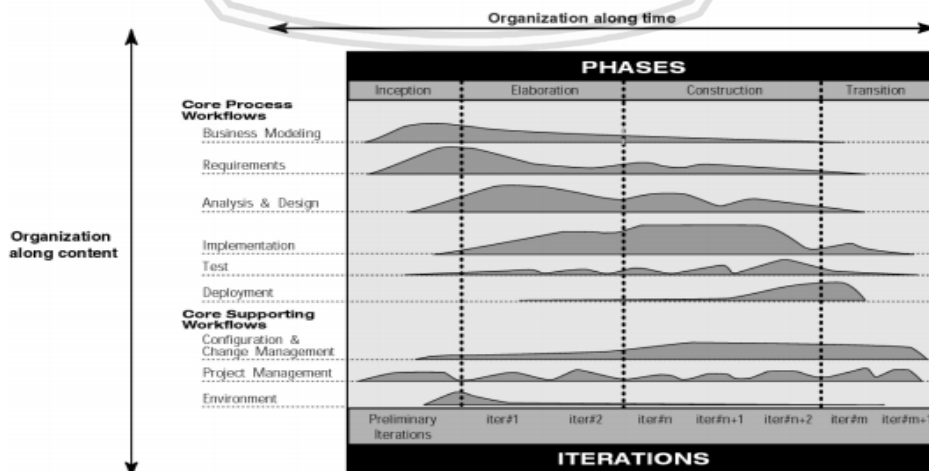
Sumber: (Pressman, 2010)

Gambar enam merupakan penggunaan diagram *sequence* pada perangkat lunak yang berfungsi untuk menggambar.

## 2.12 Rational unified Process (RUP)

Banyak *framework* SDLC yang dapat digunakan dalam pengembangan sistem informasi, salah satunya yaitu *Rational Unified Process* (RUP). RUP adalah proses rekayasa pada perangkat lunak yang menyediakan pendekatan yang disiplin dalam pengerjaan tugas-tugas dan tanggung jawab dalam organisasi pengembangan. Tujuannya yaitu untuk dapat membuat produk berkualitas tinggi dan dapat memenuhi kebutuhan pengguna, dalam jangkauan waktu dan biaya yang telah direncanakan sebelumnya. RUP dibuat untuk mendukung penggunaan *Unified Modeling Language* (UML) secara efektif (IBM, 1998).

Notasi yang digunakan pada RUP juga telah terbukti dapat menangkap dan mengelola persyaratan fungsional dan memastikan bahwa hal ini mengontrol desain, implementasi dan *testing* dari perangkat lunak sehingga dapat memenuhi kebutuhan pengguna (IBM, 1998).



Gambar 2.9 Fase pada Metode RUP

Sumber: (IBM, 1998)

Gambar 2.9 menunjukkan fase dan proses dalam penggunaan kerangka kerja RUP. Fase dan proses yang ada pada RUP dapat dilihat dari dua jenis sudut pandang, antara lain secara horizontal dan vertikal.

Terdapat empat fase pada RUP sudut pandang horizontal (IBM, 1998) yaitu:

1. Fase *Inception*

Pada fase ini, seluruh entitas eksternal akan diidentifikasi serta integrasi antara entitas didefinisikan. Hal ini juga mencakup mengidentifikasi seluruh *use-case* serta mendeskripsikan *use-case* yang signifikan. Keluaran dari fase ini yaitu *vision document*, *use-case* model awal, *business case* model awal, *project glossary* awal, rencana proyek awal, model bisnis, *risk assessment* awal, dan beberapa prototype.

2. Fase *Elaboration*

Pada fase ini domain masalah dianalisis, elemen berisiko tinggi dieliminasi, arsitektur dibuat, serta rencana proyek dibangun. Selain itu, dibangun pemahaman seluruh sistem mencakup lingkup detail persyaratan fungsional dan non fungsional. Berikut keluaran dari fase ini.

- Model *use-case*, seluruh aktor dan *use-case* telah diidentifikasi serta dideskripsikan
- Persyaratan pendukung
- Deskripsi arsitektur *software*
- *Prototype* yang dapat dieksekusi
- Daftar resiko dan *business case* yang telah direvisi
- Rencana pembangunan
- Development case yang telah diperbarui

3. Fase *Construction*

Pada fase ini, seluruh fitur dan komponen sistem dibangun dan seluruh fitur diuji.

- *Software*
- Buku panduan

4. Fase *Transition*

Pada fase ini, produk yang sudah dibangun akan ditransisikan ke pengguna. Dalam fase ini didapatkan umpan balik terhadap dari pengguna sehingga dapat digunakan sebagai catatan untuk perbaikan perangkat lunak selanjutnya.

Dari sudut pandang vertikal, proses yang dapat dilihat yaitu kegiatan yang dilakukan pada setiap fase.

1. Pemodelan bisnis

Pada kegiatan ini, proses bisnis dimodelkan untuk memastikan pemahaman dari seluruh pemangku kepentingan pada proses bisnis tersebut. Tidak semua proyek harus melakukan kegiatan ini.

2. Kebutuhan

Pada kegiatan ini bertujuan untuk mendeskripsikan apa yang sistem dapat lakukan. Kegiatan ini dilakukan pemodelan *use-case* beserta deskripsi setiap *use-case* terkait.

3. Analisis dan Desain

Pada kegiatan ini bertujuan untuk menunjukkan bagaimana sistem akan direalisasikan pada tahap implementasi. Hasil dari sistem berupa model desain dan model analisis. Pada model analisis, dilakukan analisis terhadap kebutuhan untuk mendapatkan elemen-elemen desain. Model desain terdiri dari struktur kelas dengan pendefinisian antarmuka yang merepresentasikan komponen pada saat implementasi. Juga mendefinisikan bagaimana interaksi antar kelas.

4. Implementasi

Pada kegiatan ini bertujuan untuk mengimplementasi model perancangan yang telah didefinisikan di awal menjadi sebuah produk jadi berupa perangkat lunak.

5. Testing

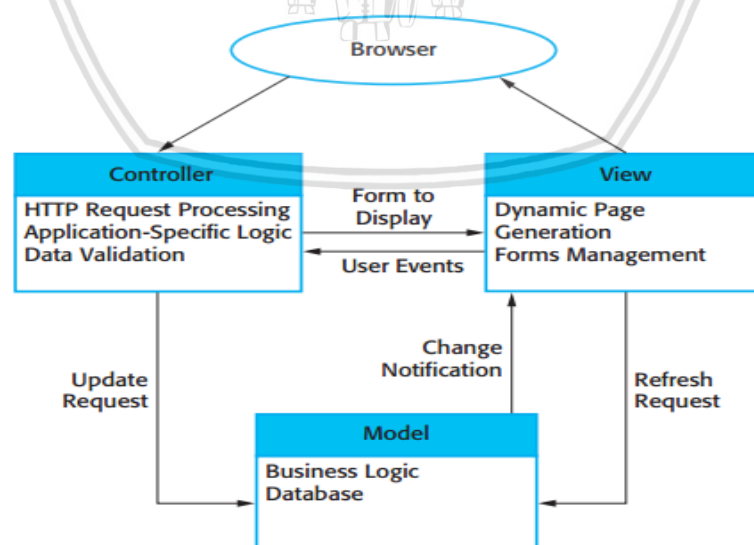
Pada kegiatan ini dilakukan pengujian terhadap perangkat lunak untuk mengetahui hasil kesesuaian berdasarkan kebutuhan yang dilakukan pada setiap akhir fase.

6. Deployment

Pada kegiatan ini produk yang telah dibangun di rilis kepada *client*.

### 2.13 Model - View – Controller

Model-View-Controller atau biasa disebut sebagai MVC adalah pola arsitektur sistem yang terbagi menjadi tiga komponen yang saling berkaitan (Sommerville, 2011). Ketiga komponen tersebut mempunyai fungsi atau kegunaan yang berbeda.



Gambar 2.10 Arsitektur berpola MVC

Sumber: (Sommerville, 2011)

Pada gambar 2.10, merupakan gambar dari aplikasi web yang menggunakan arsitektur dengan pola MVC. Komponen *model* berfungsi sebagai mengelola data pada sistem dan operasi terkait data tersebut. Komponen *view* berfungsi sebagai menampilkan informasi dengan tampilan kepada pengguna sistem. Data yang diperoleh dari komponen *model* akan memengaruhi informasi yang ditampilkan pada *view*. Komponen *controller* berfungsi sebagai mengelola interaksi yang diberikan pengguna sistem pada sistem. Sehingga komponen *controller* merupakan jembatan antara presentasi dan *database*.

## 2.14 Pengujian

Pengujian merupakan tahapan untuk memastikan perangkat lunak yang telah dikembangkan berjalan sesuai yang direncanakan serta untuk mengetahui kecacatan pada perangkat lunak. Pengujian perangkat lunak terdiri dari berbagai jenis pengujian yang mempunyai fokus yang berbeda pula (Pressman, 2010). Pengujian yang dilakukan dalam penelitian ini yaitu *black-box*, *compatibility*, dan *user acceptance testing*.

### 2.14.1 Black-box Testing

Pengujian *black-box* merupakan suatu teknik pengujian pada perangkat lunak yang berfokus pada persyaratan fungsional. Pengujian *black-box* memungkinkan pengembang perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.

Pengujian *black-box* bertujuan untuk menemukan kesalahan dalam katagori sebagai berikut :

1. Fungsi-fungsi yang telah dibuat tidak benar atau hilang.
2. Kesalahan pada *interface*.
3. Kesalahan dalam struktur data atau akses pada database eksternal.
4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminasi.

Pengujian *black-box* digunakan pada tahap akhir pengujian, karena *black-box* memperhatikan stuktur kontrol, maka perhatian berfokus pada domain informasi (Pressman, 2010).

### 2.14.2 User Acceptance Testing

*User Acceptance Testing* (UAT) merupakan pengujian yang menguji tingkat penerimaan sistem berdasarkan harapan pengguna dari sistem tersebut. Pemangku kepentingan harus memiliki andil untuk memastikan bahwa sistem yang telah dibangun telah memenuhi kebutuhan dan harapan dari pengguna akhir supaya sistem tersebut diterima dan dapat digunakan. Pengujian ini harus dapat diukur sehingga dapat diketahui sejauh mana sistem diterima oleh pengguna. Pada pengujian *User Acceptance Testing* terdapat beberapa kriteria yang dapat membantu dalam penilaian. Penerapan pada pengujian ini tidak



diharuskan semua kriteria yang ada untuk diuji. Berikut merupakan kriteria – kriteria yang ada pada pengujian UAT (Naik & Tripathy, 2008).

**Tabel 2.4 Kriteria pengujian UAT**

Kriteria	Deskripsi
<i>Functional Correctness and Completeness</i>	Berfokus pada apakah sistem yang dibangun telah sesuai dengan harapan berdasarkan pada fitur yang didefinisikan sebelumnya.
<i>Accuracy</i>	Berfokus apakah proses perhitungan atau hasil akhir dari sistem telah sesuai dengan nilai seharusnya.
<i>Data Integrity</i>	Berfokus pada performa atau kinerja sistem pada proses pengelolaan data.
<i>Data Conversion</i>	Berfokus pada performa sistem dalam menjaga ketepatan data pada saat terjadi perubahan jenis maupun bentuk dari data tersebut.
<i>Backup and Recovery</i>	Berfokus pada pencadangan data serta pemulihan data ketika terjadi suatu kejadian.
<i>Competitive Edge</i>	Berfokus pada keunggulan yang diberikan dibandingkan model lainnya.
<i>Usability</i>	Berfokus pada kemudahan yang diberikan dalam pengoperasian sistem.
<i>Performance</i>	Berfokus pada kinerja sistem.

Dari definisi di atas, dapat disimpulkan bahwa *user acceptance testing* adalah pengujian yang dilakukan oleh pengguna untuk memastikan bahwa fungsi-fungsi yang disediakan tersebut telah berjalan dengan baik dan sesuai dengan kebutuhan pengguna yang telah didefinisikan.

### **2.14.3 Compatibility**

Pengujian *compatibility* dilakukan untuk mengetahui apakah sistem dapat dioperasikan pada lingkungan yang berbeda. Pada setiap konfigurasi komputasi dapat memengaruhi proses dari aplikasi berbasis web. Perbedaan pada *browser* biasanya memberikan hasil yang berbeda seperti kecepatan pemrosesan pada sisi-klien, resolusi layar, dan kecepatan koneksi.

Beberapa permasalahan yang sering ditemukan dalam *compatibility testing* adalah lamanya saat membuka aplikasi web, *font style* dapat berubah bahkan menjadi tidak terbaca ketika menggunakan *browser* yang berbeda. Alasan penulis menggunakan *compatibility testing* yaitu untuk mengetahui terdapat masalah apa yang terjadi ketika aplikasi sistem informasi dijalankan pada lingkungan yang berbeda. (Pressman, 2010)

## 2.15 Skala Likert

Skala Likert merupakan skala tanggapan psikometri yang biasa digunakan untuk mengukur pendapat atau sikap seseorang dalam mendapatkan tingkat persetujuan dengan menggunakan serangkaian pernyataan berdasarkan definisi operasional dari peneliti (Choizes, 2018). Pada skala Likert secara umum menggunakan skala yang terdiri dari lima poin. Gambar 2.11 merupakan contoh skala Likert dengan menggunakan skala lima poin:



**Gambar 2.11 Contoh Skala Likert**

Perhitungan skala Likert diawali dengan memberikan bobot nilai pada setiap jawaban. Tabel 2.4 merupakan rincian bobot nilai jawaban. Pada tabel 2.5 merupakan jarak persentase untuk setiap jawaban.

**Tabel 2.4 Bobot Nilai**

Pilihan Jawaban	Bobot Nilai
A = Sangat Setuju	5
B = Setuju	4
C = Netral	3
D = Tidak Setuju	2
E = Sangat Tidak Setuju	1

Sumber: (Choizes, 2018)

**Tabel 2.5 Persentase nilai**

Interval Persentase	Interpretasi Skor
80 sampai 100%	Sangat Setuju
60 sampai 79,99%	Setuju
40 sampai 59,99%	Netral
20 sampai 39,99%	Tidak Setuju
0 sampai 19,99%	Sangat Tidak Setuju

Sumber: (Choizes, 2018)

Setelah bobot nilai dan persentase nilai telah ditentukan, selanjutnya adalah mengkalikan jumlah jawaban dengan bobot nilainya. Untuk perhitungannya akan dijelaskan pada Rumus berikut:

1. Jawaban Sangat Setuju (A)  $= n \times 5$
2. Jawaban Setuju (B)  $= n \times 4$
3. Jawaban Netral (C)  $= n \times 3$
4. Jawaban Tidak Setuju (D)  $= n \times 2$
5. Jawaban Sangat Tidak Setuju (E)  $= n \times 1$

$$\text{Total Nilai} = (n \times 5) + (n \times 4) + (n \times 3) + (n \times 2) + (n \times 1) \quad (2.3)$$

$n$  merupakan jumlah dari responden yang menjawab. Perhitungan selanjutnya adalah mencari nilai terendah ( $Y$ ) dan nilai tertinggi dari hasil skala *likert* tersebut. Rumus yang digunakan dijelaskan pada Rumus 2.4 dan Rumus 2.5, di mana  $N1$  merupakan nilai tertinggi skala Likert,  $N2$  merupakan nilai terendah skala Likert,  $n$  merupakan jumlah responden, dan  $U$  merupakan jumlah uji kasus.

$$Y = N1 \times n \times U \quad (2.4)$$

$$Y = N1 \times n \times U \quad (2.5)$$

Sedangkan untuk menghitung hasil dari UAT dengan menggunakan skala Likert dijelaskan pada Rumus 2.6 berikut.

$$\text{Rumus Index \%} = (\text{Total Nilai} \div Y) \times 100 \quad (2.6)$$

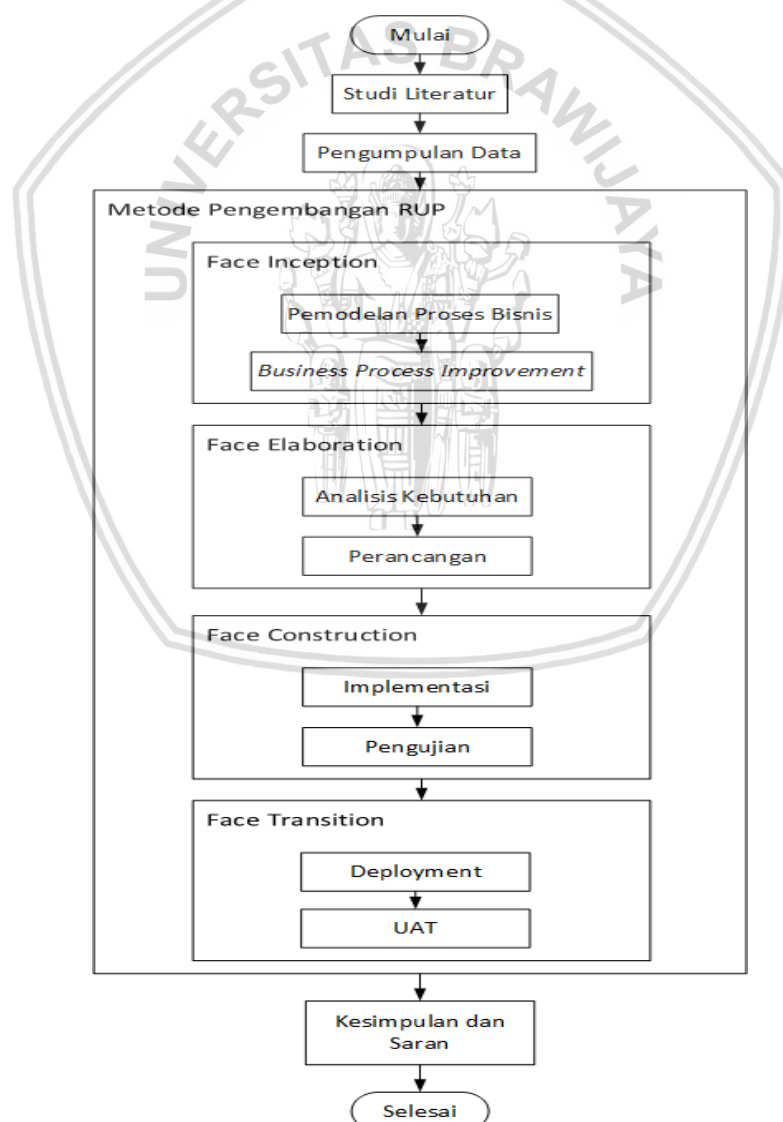
## BAB 3 METODOLOGI PENELITIAN

### 3.1 Tempat dan Waktu Penelitian

Penelitian dilakukan pada bulan Februari 2018 hingga Agustus 2018. Penelitian ini bertempat di PT Mahkota Mitra Sentosa yang beralamatkan di Jln. R.A. Kartini 121 Kayenlor Ds. Plemahan Kab. Kediri untuk melakukan studi lapangan, dan Laboratorium Sistem Informasi Fakultas Ilmu Komputer Universitas Brawijaya Malang untuk melakukan penelitian.

### 3.2 Langkah-Langkah Penelitian

Penelitian perancangan sistem informasi penggajian ini merupakan tipe penelitian implementatif. Penelitian ini dilakukan untuk PT Mahkota Mitra Sentosa. Langkah-langkah dalam melakukan penelitian disajikan dalam Gambar 3.1



Gambar 3.3.1 Alur Penelitian

### 3.2.1 Studi Literatur

Tahap ini merupakan tahap pengumpulan literatur dan referensi dari jurnal, buku, *e-book*, dan penelitian yang telah dilakukan sebelumnya terkait dengan topik permasalahan untuk mendapatkan teori dasar dan pendukung penelitian ini. Hal ini bertujuan dapat meningkatkan pemahaman serta pengetahuan peneliti terhadap permasalahan yang diangkat. Pada tahap ini didapatkan beberapa teori pendukung penelitian meliputi:

- a. Penelitian yang berkaitan sebelumnya
- b. Sistem informasi
- c. *Unified Modeling Language* (UML)
- d. *Rational Unified Process* (RUP)
- e. *Framework Codeigniter*

### 3.2.2 Pengumpulan Data

Tahap pengumpulan data termasuk dalam fase *inception* pada RUP. Tahap di mana peneliti akan mengumpulkan informasi mengenai proses bisnis dan informasi yang dibutuhkan untuk dilakukan analisa sistem yang dikembangkan. Data yang diperoleh dari tahap ini juga berpengaruh pada hasil penelitian. Metode yang digunakan untuk melakukan pengumpulan data pada penelitian ini antara lain:

#### 3.2.2.1 Observasi

Observasi dilakukan untuk mendapatkan gambaran secara langsung mengenai tingkah laku terkait proses bisnis penggajian karyawan PT Mahkota Mitra Sentosa. Observasi digunakan untuk mendapatkan gambaran visual secara langsung untuk memperkuat wawasan berdasarkan tingkah laku seseorang dalam penelitian ini.

#### 3.2.2.2 Wawancara

Wawancara merupakan salah satu metode yang dilakukan untuk mendapatkan atau mengumpulkan data yang dibutuhkan dalam penelitian ini. Data yang telah terkumpul akan dijadikan sebuah acuan dalam mengembangkan sebuah sistem penggajian karyawan. Wawancara dilakukan dengan pihak terkait yang mempunyai hubungan dengan penggajian karyawan. Wawancara akan dilakukan kepada pimpinan PT Mahkota Mitra Sentosa dan bagian administrasi. Wawancara tersebut dilakukan dengan menggali informasi terkait permasalahan yang ada, proses bisnis yang berjalan, kebutuhan sistem, dan mendapatkan data-data lainnya yang dapat mendukung penelitian ini

### 3.2.3 Pemodelan Proses Bisnis

Pada tahap pemodelan proses bisnis dalam metode *Rational Unified Proses* (RUP) lebih fokus dalam fase *inception*. Pada tahap ini melakukan pemodelan proses bisnis yang telah berjalan terkait bisnis proses penggajian karyawan, bisnis proses sewa bus, bisnis proses membuat laporan unit dan laporan global



berdasarkan data yang telah dikumpulkan tersebut. Pemodelan bisnis proses tersebut menggunakan perangkat lunak bizagi.

### 3.2.4 Business Process Improvement

Pada tahap *business process improvement* dalam metode *Rational Unified Proses* (RUP) lebih fokus dalam fase *inception*. Pada tahap ini dilakukan pemodelan proses bisnis usulan. Dalam tahap ini juga dijabarkan mengenai perubahan yang terjadi pada bisnis proses yang sedang berjalan. Selain itu, dilakukan perbandingan waktu lama berjalannya suatu proses dengan menggunakan simulai dari bizagi. Bisnis proses yang dibahas pada tahap ini yaitu bisnis proses penggajian karyawan, bisnis proses sewa bus, bisnis proses membuat laporan unit dan laporan global.

### 3.2.5 Analisis Kebutuhan

Tahap ini merupakan tahap analisa kebutuhan. Pada tahap analisa dalam metode *Rational Unified Proses* (RUP) lebih fokus dalam fase *inception* dan fase *elaboration*. Tahap analisa kebutuhan dilakukan untuk mengetahui dan mendapatkan kebutuhan pengguna terhadap sistem yang akan dikembangkan dari hasil pengumpulan data. Dalam fase *inception* peneliti membuat *vision document*. Pada fase *elaboration* peneliti mendefinisikan kebutuhan pengguna tersebut berupa kebutuhan fungsional dan kebutuhan non fungsional, pemodelan *use-case*, membuat *scenario* pada masing-masing *use-case*, dan memodelkan *activity diagram*.

### 3.2.6 Perancangan Sistem

Pada tahap perancangan sistem ini masuk dalam fase *elaboration* pada fase RUP. Pada tahap ini dilakukan perancangan sistem yaitu dengan pembuatan *sequence diagram*, *class diagram*, dan *physical data model* dari hasil proses bisnis yang telah dimodelkan serta kebutuhan fungsional dan kebutuhan non fungsional yang telah dianalisis pada langkah sebelumnya yaitu pada analisis kebutuhan sistem. Setelah diagram telah dibuat langkah selanjutnya yaitu membuat perancangan *User Interface* (UI) atau tampilan yang digunakan dalam sistem mengenai tata letak dari setiap fungsi dan hal lainnya yang berhubungan dengan tampilan sehingga dapat terlihat bagaimana sistem ini akan berjalan ketika nantinya sudah ditransisikan pada PT Mahkota Mitra Sentosa. Kemudian dibuat perancangan algoritme.

### 3.2.7 Implementasi

Pada tahap implementasi masuk ke dalam fase *construction* pada RUP. Pada tahap ini seluruh perancangan desain diagram yang telah dibuat sebelumnya akan diimplementasikan ke dalam bentuk kode program menggunakan metode *object oriented programming* (OOP) dengan bahasa pemrograman PHP dan SQL untuk mengimplementasikan *physical data model*. Serta menggunakan bahasa pemrograman HTML, CSS, Javascript, dan aplikasi xampp sebagai *web service localhost*. Hasil dari implementasi ini merupakan sebuah sistem yang sesuai

berdasarkan rancangan-rancangan yang telah dibuat sebelumnya dan sebagai landasan untuk melakukan pengujian.

### 3.2.8 Pengujian

Tahap ini merupakan proses pengujian terhadap sistem yang telah diimplementasikan sebelumnya. Pada tahap pengujian dalam metode *Rational Unified Proses* (RUP) lebih fokus dalam fase *construction*. Tujuan pada tahap pengujian ini adalah untuk memastikan bahwa semua fungsi sistem telah berjalan semestinya dan mencari kesalahan yang mungkin terjadi pada sistem (*error*).

Metode yang digunakan dalam tahap pengujian ini adalah menggunakan metode pengujian *black-box* dan *compatibility testing*. *Black-box testing* bertujuan untuk memastikan dan memeriksa bahwa sistem telah memenuhi semua fungsional yang telah didefinisikan. Pengujian *black-box* dilakukan dengan menjalankan masing-masing fungsi yang akan diuji coba dengan memasukkan data yang sesuai maupun data yang tidak sesuai dengan masukkan untuk melihat bagaimana respon sistem dari data yang dimasukkan. Sedangkan *compatibility testing* bertujuan untuk mengetahui bagaimana respon dari sistem apabila dijalankan pada lingkungan sistem yang berbeda. Lingkungan sistem yang dimaksud yaitu pada *browser* yang berbeda-beda. *Compatibility testing* ini dilakukan dengan menggunakan bantuan perangkat lunak yang bernama *SortSite*. *SortSite* akan melakukan pengujian *compatibility* secara otomatis terhadap sistem yang dibuat dengan cara memasukkan alamat url dari sistem yang dibangun.

### 3.2.9 Deployment

Pada tahap *deployment* masuk ke dalam fase *transition* pada RUP. Tahap ini yaitu tahap di mana sistem yang telah dibangun dan telah dilakukan pengujian menggunakan *black-box* dan *compatibility* kemudian diimplementasikan pada lingkungan sistem PT Mahkota Mitra Sentosa.

Proses implementasi yang dilakukan di PT Mahkota Mitra Sentosa bertujuan untuk memberikan pengalaman interaksi pengguna terhadap sistem, pengenalan fitur, dan interaksi yang terjadi di dalam sistem hingga menjadi respon yang diberikan oleh sistem terhadap perlakuan pengguna. Dengan proses tersebut, pengguna diberikan kesempatan untuk menyampaikan pendapatnya mengenai sistem yang telah dibangun, apakah sudah baik dan dapat segera dirilis ataukah masih terdapat kekurangan atau masukan agar sistem menjadi lebih baik dan dapat memenuhi kebutuhan dari pengguna itu sendiri sehingga dapat dijadikan sebagai evaluasi. Hasil evaluasi ini menjadi dasar dari proses pendefinisian ulang, penambahan kebutuhan sistem, perbaikan sistem apabila terdapat kekurangan atau fungsi yang tidak sesuai dengan kebutuhan pengguna.

### 3.2.10 User Acceptance Testing (UAT)

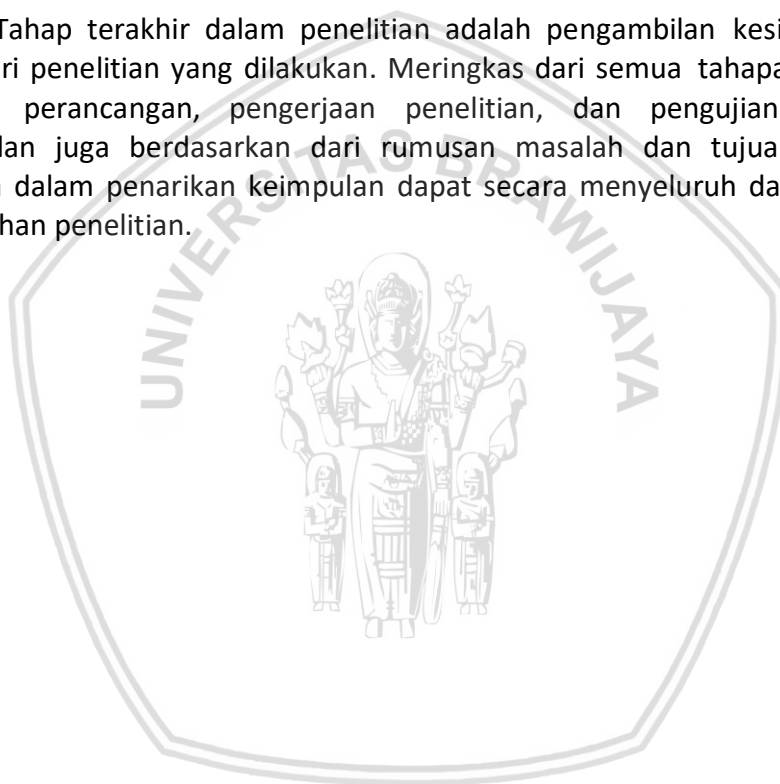
Tahap *User Acceptance Testing* (UAT) masuk dalam fase *transition* pada RUP. Tahapan ini bersamaan dengan tahap *deployment* yang bertujuan untuk

melakukan evaluasi terhadap sistem yang telah dibangun. Evaluasi ini untuk mengetahui apakah sistem telah memenuhi kriteria dan kebutuhan sistem pengguna atau belum.

Pada *User Acceptance Testing (UAT)* ini dibuat sebuah *scenario* uji terhadap fungsi dari sistem yang telah dibuat dalam bentuk kuesioner. Pengujian ini dilakukan oleh pimpinan dari PT Mahkota Mitra Sentosa dan bagian administrasi karena semua aktor tersebut yang akan menggunakan sistem informasi. Setelah pengguna sistem menjalankan *scenario* yang telah dibuat, pengguna akan mengisi kuesioner apakah *scenario* yang dijalankan sesuai dengan pernyataan yang tertulis atau tidak.

### 3.2.11 Kesimpulan dan Saran

Tahap terakhir dalam penelitian adalah pengambilan kesimpulan dan saran dari penelitian yang dilakukan. Meringkas dari semua tahapan penelitian meliputi perancangan, pengerjaan penelitian, dan pengujian. Penarikan kesimpulan juga berdasarkan dari rumusan masalah dan tujuan penelitian sehingga dalam penarikan kesimpulan dapat secara menyeluruh dan mencakup keseluruhan penelitian.



## BAB 4 ANALISIS KEBUTUHAN

### 4.1 Pemodelan Proses Bisnis

Pada sub bab ini dilakukan pemodelan proses bisnis untuk memodelkan serangkaian aktivitas yang dilakukan oleh fungsi-fungsi bisnis pada PT Mahkota Mitra Sentosa untuk memenuhi kebutuhan pengguna bisnis. Penggalian informasi mengenai proses bisnis yang akan dimodelkan didapatkan dengan cara melakukan wawancara dengan bagian administrasi dan pimpinan PT Mahkota Mitra Sentosa mengenai alur proses bisnis pada PT Mahkota Mitra Sentosa dan dilakukan observasi pada lokasi studi kasus penelitian. Dalam penelitian ini, proses bisnis yang akan dilakukan perbaikan, antara lain:

1. Proses bisnis sewa bus
2. Proses bisnis membuat slip gaji
3. Proses bisnis membuat laporan unit
4. Proses bisnis membuat laporan global

#### 4.1.1 Proses Bisnis *As-Is*

Dalam penelitian ini proses bisnis *as-is* dapat diidentifikasi berdasarkan keadaan proses bisnis yang berjalan pada lokasi studi kasus sebelum solusi yang diusulkan dalam penelitian ini diberikan. Analisis dan pemodelan terhadap proses bisnis *as-is* dilakukan agar peneliti dapat mengidentifikasi kemungkinan perbaikan atau bahkan peningkatan terhadap proses bisnis melalui solusi yang ditawarkan dalam penelitian ini.

##### 4.1.1.1 Proses Bisnis Sewa Bus

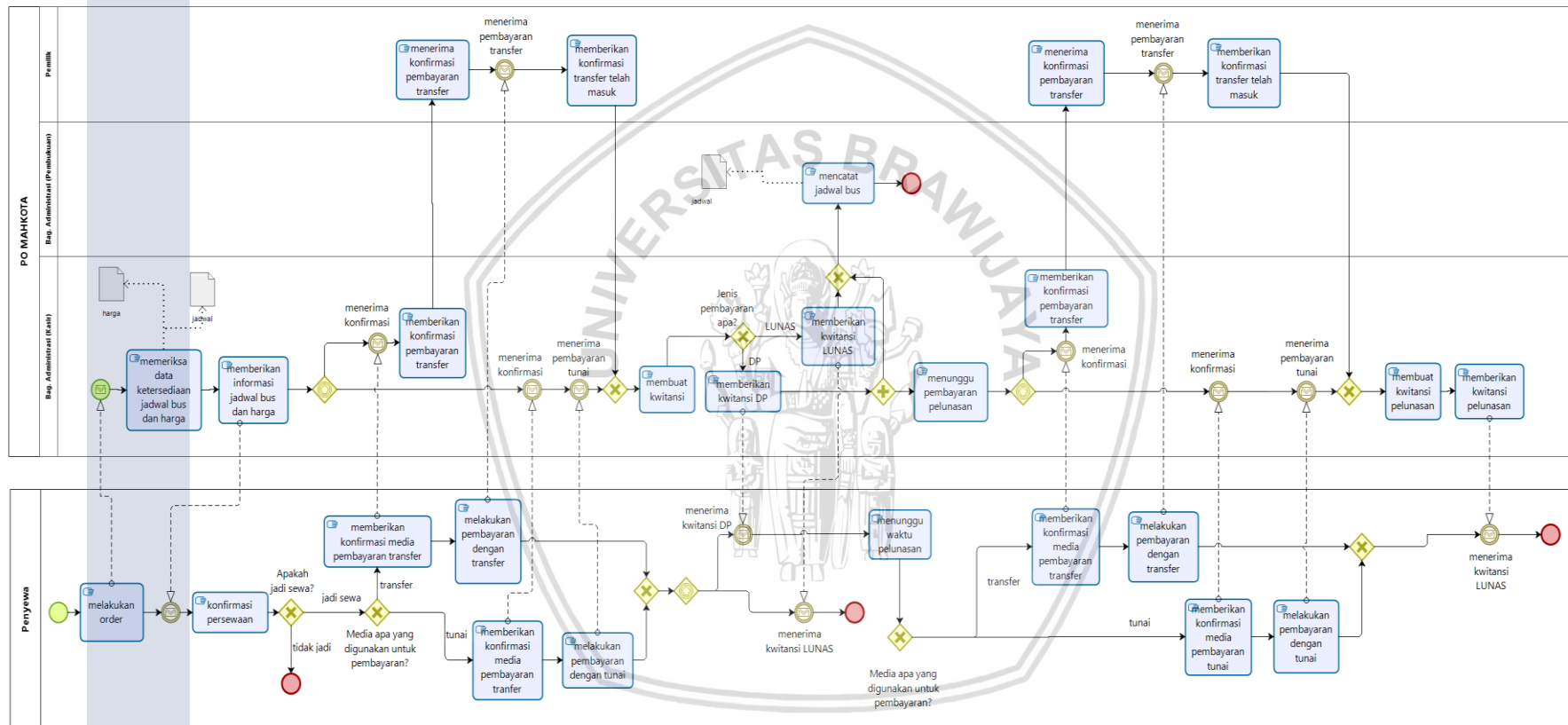
Dalam Gambar 4.1 menjelaskan rangkaian aktivitas sewa bus yang dilakukan oleh fungsi bisnis bagian administrasi dan pimpinan PT Mahkota Mitra Sentosa. Proses bisnis dimulai ketika penyewa sebagai pelanggan bisnis eksternal PT Mahkota Mitra Sentosa ingin melakukan sewa bus dengan memulai aktivitas melakukan order melalui bagian administrasi. Kemudian bagian administrasi akan melakukan aktivitas memeriksa data ketersediaan jadwal bus dan harga. Bagian administrasi akan memberikan informasi mengenai jadwal bus dan harga.

Penyewa akan melakukan konfirmasi sewa apakah jadi sewa atau tidak setelah penyewa menerima informasi mengenai jadwal bus dan harga. Ketika penyewa jadi sewa, maka penyewa akan melakukan pembayaran sewa dengan DP atau dengan lunas sesuai dengan informasi harga yang telah diberikan oleh bagian administrasi. Dalam melakukan pembayaran, penyewa dapat melakukan pembayaran transfer atau dengan tunai dengan melakukan konfirmasi terlebih dahulu kepada bagian administrasi. Ketika penyewa melakukan pembayaran dengan transfer, bagian administrasi akan melakukan konfirmasi transfer pada komaris dan setelah komaris menerima sejumlah pembayaran akan melakukan konfirmasi kepada bagian administrasi bahwa pembayaran telah diterima sesuai. Bila pembayaran dilakukan dengan metode tunai, bagian administrasi yang akan

menerima pembayaran tersebut. Setelah itu bagian administasi akan membuat kuitansi DP atau lunas sesuai dengan pembayaran yang telah dilakukan. Kemudian memberikan kuitansi tersebut kepada penyewa. Bagian administasi melakukan pencatatan di buku jadwal sesuai dengan jadwal yang telah disewa. Ketika penyewa melakukan sewa dengan DP, maka penyewa harus melakukan pelunasan 2 hari sebelum keberangkatan. Pembayaran pelunasan pun juga sama seperti DP dapat dilakukan dengan transfer atau tunai di kantor. Setelah pembayaran pelunasan selesai penyewa akan menerima kuitansi pelunasan.



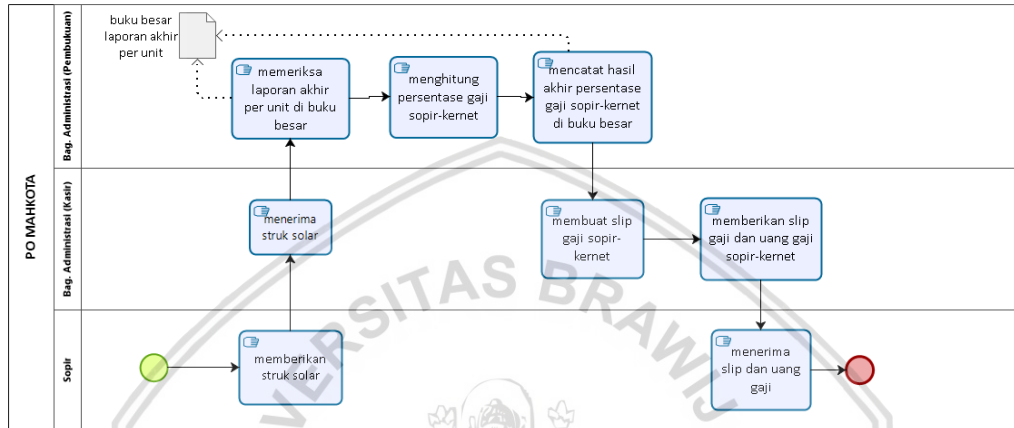




Gambar 4.1 Proses Bisnis Sewa Bus

#### 4.1.1.2 Proses Bisnis Membuat Slip Gaji

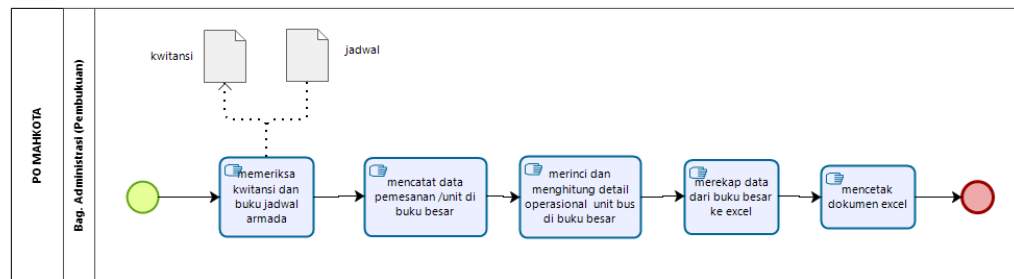
Dalam Gambar 4.2 menjelaskan rangkaian aktivitas membuat slip gaji yang dilakukan oleh fungsi bisnis bagian administrasi dan sopir. Proses bisnis dimulai ketika sopir memberikan struk solar kepada bagian administrasi. Kemudian bagian administrasi memeriksa laporan unit di buku besar. Bagian administrasi menghitung prosentase gaji dari biaya sewa dan mencatat di laporan unit dalam buku besar. Setelah itu bagian administrasi membuat slip gaji. Kemudian memberikan slip gaji beserta gaji sesuai dengan rincian pada slip gaji.



Gambar 4.2 Proses Bisnis Membuat Slip Gaji

#### 4.1.1.3 Proses Bisnis Membuat Laporan Unit

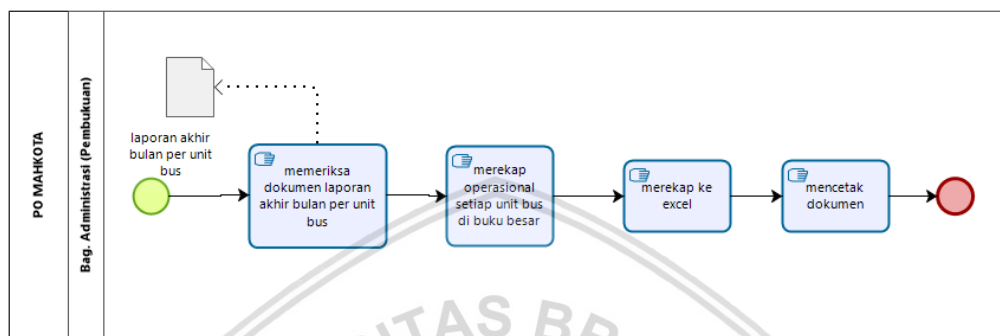
Dalam Gambar 4.3 menjelaskan rangkaian aktivitas membuat laporan unit yang dilakukan oleh fungsi bisnis bagian administrasi. Proses bisnis dimulai ketika bagian administrasi melakukan aktivitas memeriksa kuitansi dan buku jadwal. Kemudian bagian administrasi mencatat data persewaan yang telah terjadi dan merinci serta menghitung detail operasional pada buku besar. Setelah itu bagian administrasi merekap data tersebut pada dokumen *excel* dan mencetaknya.



Gambar 4.3 Proses Bisnis Membuat Laporan Unit

#### 4.1.1.4 Proses Bisnis Membuat Laporan Global

Dalam Gambar 4.4 menjelaskan rangkaian aktivitas membuat laporan global yang dilakukan oleh fungsi bisnis bagian administrasi. Proses bisnis dimulai ketika bagian administrasi melakukan aktivitas memeriksa laporan unit. Kemudian bagian administrasi merekap total operasional setiap unit pada buku besar. Setelah itu bagian administrasi merekap data tersebut pada dokumen *excel* dan mencetaknya.



Powered by  
bizagi  
Modeler

Gambar 4.4 Proses Bisnis Membuat Laporan Global

## 4.2 Business Process Improvement

Pada sub bab *business process improvement* ini akan dilakukan pemodelan proses bisnis perbaikan. Selain pemodelan proses bisnis perbaikan juga dijabarkan mengenai perbandingan dengan proses bisnis yang sedang berjalan. Proses bisnis yang diperbaiki yaitu proses bisnis sewa bus, proses bisnis membuat slip gaji, proses bisnis mencetak laporan unit, proses bisnis mencetak laporan global.

### 4.2.1 Proses Bisnis To-Be

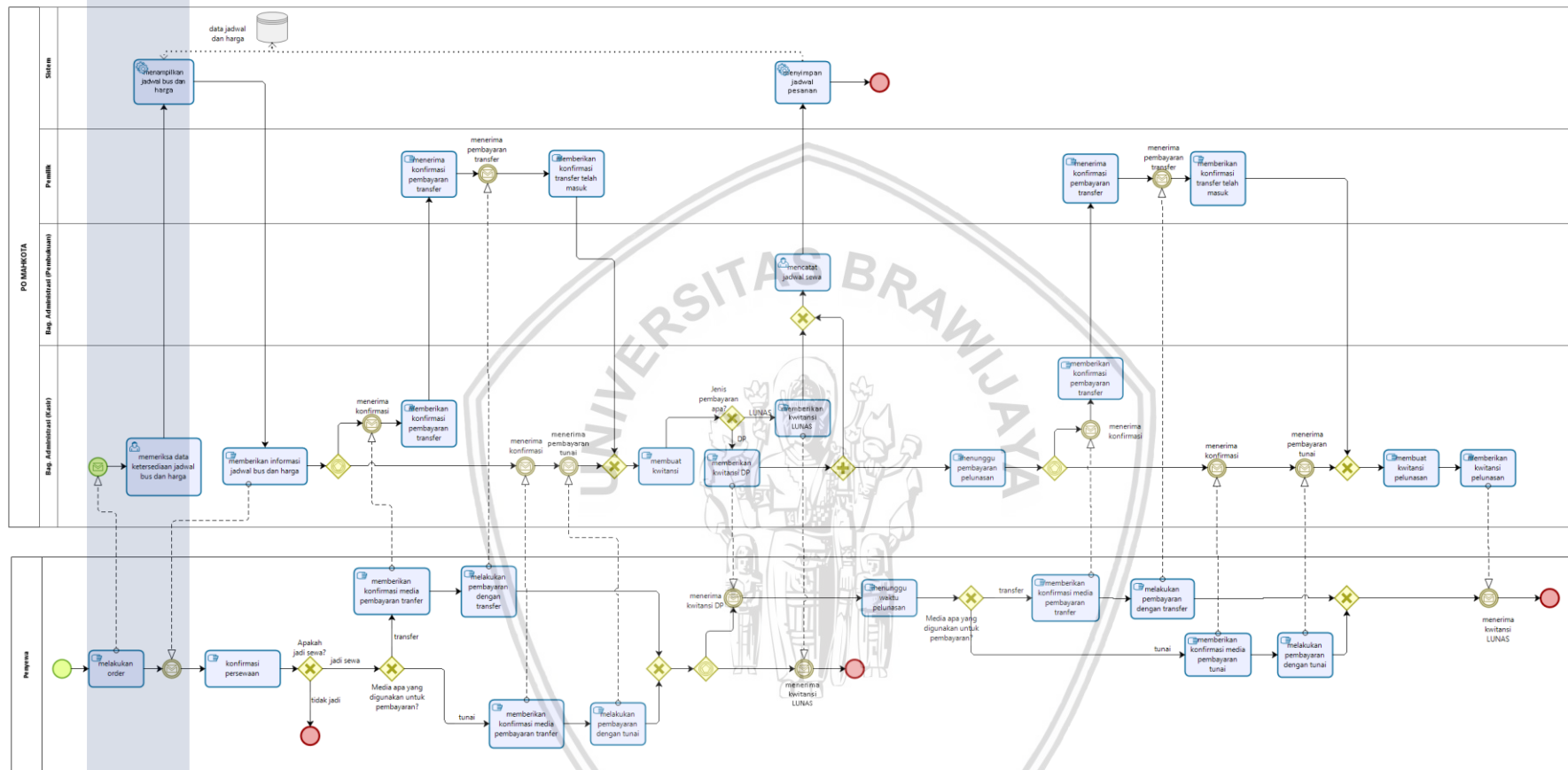
Proses bisnis *to-be* yang dimodelkan pada penelitian ini merupakan proses bisnis usulan yang ditawarkan dalam penelitian ini kepada pemangku kepentingan. Proses bisnis *to-be* diidentifikasi berdasarkan hasil analisis terhadap proses bisnis *as-is*. Dalam penelitian ini, analisis proses bisnis *to-be* dilakukan dengan melakukan penambahan dan penghapusan pada beberapa aktivitas. Selanjutnya, proses bisnis *to-be* dimodelkan untuk menggambarkan perubahan atau tambahan yang diusulkan melalui penelitian ini.

#### 4.2.1.1 Proses Bisnis Sewa Bus

Dalam Gambar 4.5 menjelaskan serangkaian aktivitas sewa bus yang dilakukan oleh fungsi bisnis bagian administrasi dan komisaris PT Mahkota Mitra Sentosa. Proses bisnis dimulai ketika penyewa sebagai pelanggan bisnis eksternal PT Mahkota Mitra Sentosa ingin melakukan sewa bus dengan memulai aktivitas melakukan order melalui bagian administrasi. Kemudian bagian administrasi akan melakukan aktivitas memeriksa data ketersediaan jadwal bus

dan harga pada sistem informasi . Bagian administasi akan memberikan informasi mengenai jadwal bus dan harga.

Penyewa akan melakukan konfirmasi sewa apakah jadi sewa atau tidak setelah penyewa menerima informasi mengenai jadwal bus dan harga. Ketika penyewa jadi sewa, maka penyewa akan melakukan pembayaran sewa dengan DP atau dengan lunas sesuai dengan informasi harga yang telah diberikan oleh bagian administasi. Dalam melakukan pembayaran, penyewa dapat melakukan pembayaran transfer atau dengan tunai dengan melakukan konfirmasi terlebih dahulu kepada bagian administasi. Ketika penyewa melakukan pembayaran dengan transfer, bagian administrasi akan melakukan konfirmasi transfer pada komisariss dan setelah komisariss menerima sejumlah pembayaran akan melakukan konfirmasi kepada bagian administasi bahwa pembayaran telah diterima sesuai. Bila pembayaran dilakukan dengan metode tunai, bagian administasi yang akan menerima pembayaran tersebut. Setelah itu bagian administasi akan membuatkan kuitansi DP atau lunas sesuai dengan pembayaran yang telah dilakukan. Kemudian memberikan kuitansi tersebut kepada penyewa. Bagian administasi melakukan pencatatan pada sistem informasi sesuai dengan jadwal yang telah disewa. Ketika penyewa melakukan sewa dengan DP, maka penyewa harus melakukan pelunasan dua hari sebelum keberangkatan. Pembayaran pelunasan pun juga sama seperti DP dapat dilakukan dengan transfer atau tunai di kantor. Setelah pembayaran pelunasan selesai penyewa akan menerima kuitansi pelunasan. Perubahan aktivitas pada bisnis proses sewa bus dijelaskan pada Tabel 4.1.



Gambar 4.5 Proses Bisnis Sewa Bus



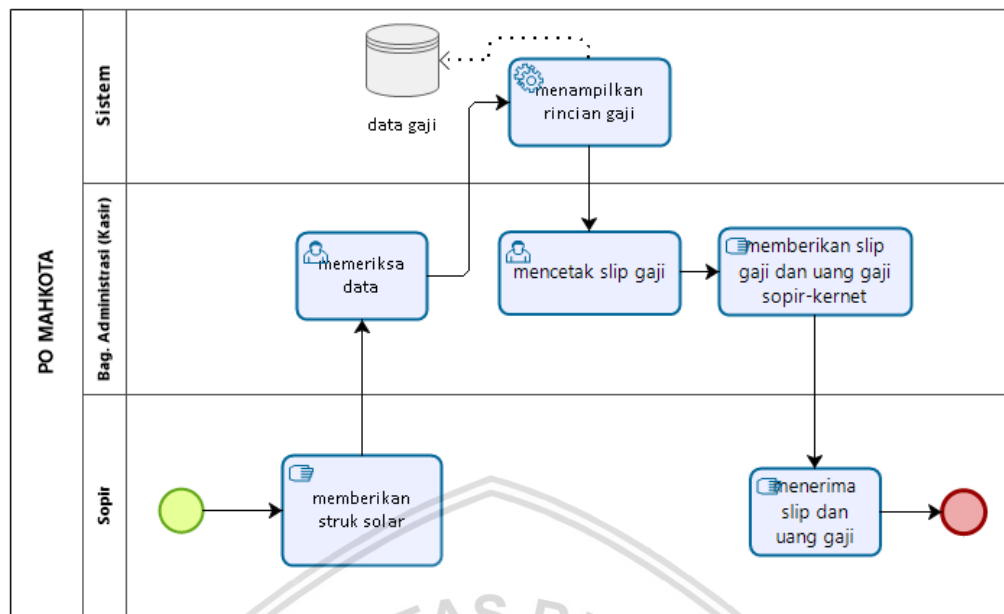
**Tabel 4.1 Perubahan Aktivitas Proses Bisnis Sewa Bus**

No.	Unit bisnis	Proses bisnis as-is	Proses bisnis to-be	Keterangan
1.	Bagian Administrasi (kasir)	Memeriksa harga dan ketersediaan bus pada buku besar	Memeriksa harga dan ketersediaan bus pada sistem	Diubah
2.	Bagian Administrasi (pembukuan)	Mencatat jadwal sewa pada buku	Mencatat jadwal sewa pada sistem	Diubah
3.	Sistem	-	Menampilkan informasi harga dan ketersediaan bus	Ditambah
4.	Sistem	-	Menyimpan jadwal sewa	Ditambah

Pada tabel 4.1 dijelaskan bahwa terdapat proses bisnis yang diganti dan ditambah. Proses bisnis memeriksa harga dan ketersediaan bus pada buku besar diganti dengan memeriksa harga dan ketersediaan bus pada sistem dan mencatat jadwal sewa pada buku diganti dengan mencatat jadwal sewa pada sistem. Kemudian proses bisnis yang ditambah adalah menampilkan informasi harga dan ketersediaan bus serta menyimpan jadwal sewa yang dilakukan oleh unit bisnis sistem.

#### **4.2.1.2 Proses Bisnis Membuat Slip Gaji**

Dalam Gambar 4.6 menjelaskan rangkaian aktivitas membuat slip gaji yang dilakukan oleh fungsi bisnis bagian administrasi dan sopir. Proses bisnis dimulai ketika sopir memberikan struk solar kepada bagian administrasi. Kemudian bagian administrasi memeriksa data pada sistem informasi. Sistem informasi akan menampilkan detail rincian gaji. Setelah itu bagian administrasi mencetak slip gaji. Kemudian memberikan slip gaji beserta gaji sesuai dengan rincian pada slip gaji. Perubahan aktivitas pada bisnis proses sewa bus dijelaskan pada Tabel 4.2.



Powered by  
**bizagi**  
Modeler

**Gambar 4.6 Proses Bisnis Membuat Slip Gaji**

**Tabel 4.2 Proses Bisnis Membuat Slip Gaji**

No.	Unit bisnis	Proses bisnis as-is	Proses bisnis to-be	Keterangan
1.	Bagian Administrasi (Pembukuan)	Memeriksa laporan akhir per unit di buku besar	-	Dihapus
2.	Bagian Administrasi (Pembukuan)	Menghitung persentase gaji sopir-kernet	-	Dihapus
3.	Bagian Administrasi (Pembukuan)	Mencatat hasil akhir persentase gaji sopir-kernet di buku besar	-	Dihapus

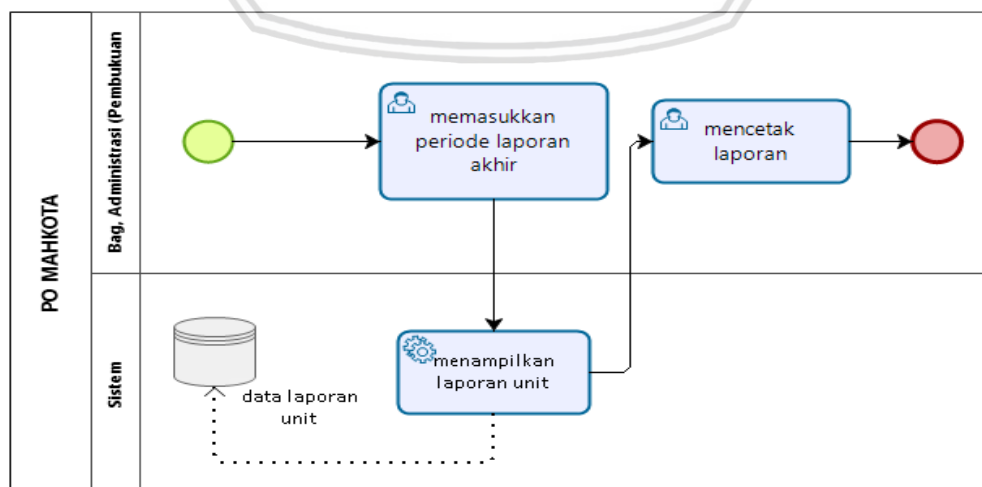
**Tabel 4.2 Proses Bisnis Membuat Slip Gaji**

No.	Unit bisnis	Proses bisnis as-is	Proses bisnis to-be	Keterangan
4.	Bagian Administrasi (Kasir)	Membuat slip gaji sopir-kernet	Mencetak slip gaji	Diubah
5.	Bagian Administrasi (Kasir)	-	Memeriksa data	Ditambah
6.	Sistem	-	Menampilkan rincian gaji	Ditambah

Pada tabel 4.2 dijelaskan bahwa terdapat proses bisnis yang diganti, dihapus, dan ditambah. Terdapat tiga proses bisnis yang dilakukan oleh bagian administrasi pembukuan dihapus. Proses bisnis pada bagian administrasi kasir yang membuat slip-gaji sopir-kernet diganti dengan mencetak slip gaji. Kemudian ada dua proses bisnis yang ditambahkan dalam to-be yaitu memeriksa data yang dilakukan oleh kasir dan menampilkan rincian gaji oleh sistem.

#### 4.2.1.3 Proses Bisnis Membuat Laporan Unit

Dalam Gambar 4.7 menjelaskan rangkaian aktivitas membuat laporan unit yang dilakukan oleh fungsi bisnis bagian administrasi. Proses bisnis dimulai ketika bagian administasi memasukkan periode laporan. Kemudian sistem informasi akan menampilkan laporan unit sesuai dengan periode yang telah dimasukkan bagian administrasi lalu mencetaknya. Perubahan aktivitas pada bisnis proses sewa bus dijelaskan pada Tabel 4.3.



**Gambar 4.7 Proses Bisnis Membuat Laporan Unit**

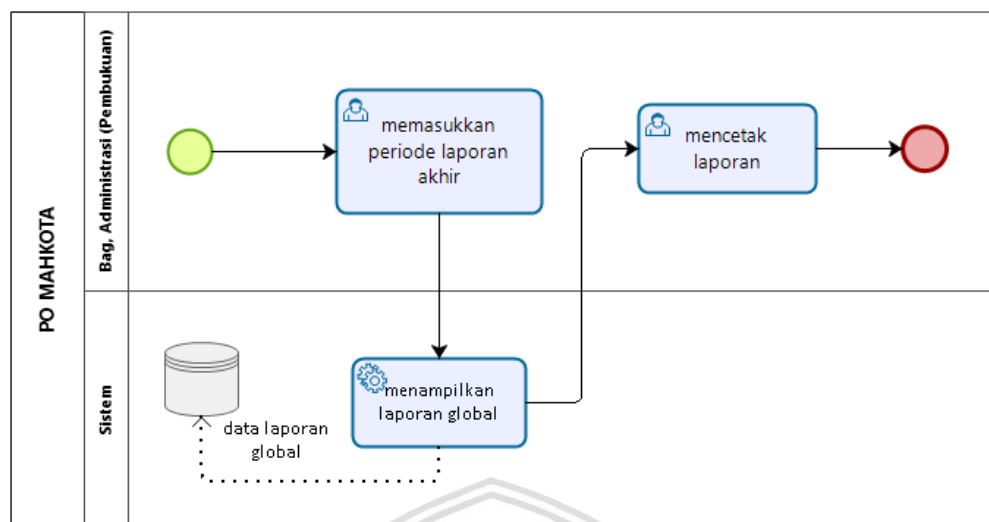
Tabel 4.3 Proses Bisnis Membuat Laporan Unit

No.	Unit bisnis	Proses bisnis as-is	Proses bisnis to-be	Keterangan
1.	Bagian Administrasi (Pembukuan)	Memeriksa kuitansi dan buku jadwal armada	Memasukkan periode laporan akhir pada sistem	Diubah
2.	Bagian Administrasi (Pembukuan)	Mencatat data pemesanan /unit di buku besar	-	Dihapus
3.	Bagian Administrasi (Pembukuan)	Merinci dan menghitung detail operasional unit bus di buku besar	-	Dihapus
4.	Bagian Administrasi (Pembukuan)	Merekap data dari buku besar ke excel	-	Dihapus
5.	Sistem	-	Menampilkan laporan unit	Ditambah

Pada tabel 4.3 dijelaskan bahwa terdapat proses bisnis yang diganti, dihapus, dan ditambah. Proses bisnis memeriksa kuitansi dan buku jadwal armada diganti dengan proses bisnis memasukkan priode laporan akhir pada sistem. Kemudian terdapat tiga proses bisnis yang dihapus pada *to-be*. Proses bisnis menampilkan laporan unit ditambahkan pada proses bisnis *to-be* yang dilakukan oleh sistem.

#### 4.2.1.4 Proses Bisnis Membuat Laporan Global

Dalam Gambar 4.8 menjelaskan rangkaian aktivitas membuat laporan global yang dilakukan oleh fungsi bisnis bagian administrasi. Proses bisnis dimulai ketika bagian administasi memasukkan periode laporan. Kemudian sistem informasi akan menampilkan laporan global sesuai dengan perionde yang telah dimasukkan bagian administrasi lalu mencetaknya. Perubahan aktivitas pada bisnis proses sewa bus dijelaskan pada Tabel 4.4.



**Gambar 4.8 Proses Bisnis Membuat Laporan Global**

**Tabel 4.4 Proses Bisnis Membuat Laporan Global**

No.	Unit bisnis	Proses bisnis as-is	Proses bisnis to-be	Keterangan
1.	Bagian Administrasi (Pembukuan)	Memeriksa dokumen laporan akhir bulan per unit bus	Memasukkan periode laporan akhir pada sistem	Diubah
2.	Bagian Administrasi (Pembukuan)	Merekap operasional setiap unit bus di buku besar	-	Dihapus
3.	Bagian Administrasi (Pembukuan)	Merekap data dari buku besar ke <i>excel</i>	-	Dihapus
4.	Sistem	-	Menampilkan laporan unit	Ditambah

Pada tabel 4.4 dijelaskan bahwa terdapat proses bisnis yang diganti, dihapus, dan ditambah. Proses bisnis yang diganti adalah proses bisnis memeriksa dokumen laporan akhir bulan per unit bus yang dilakukan oleh bagian administrasi pembukuan. Kemudian proses bisnis yang dihapus adalah merekap operasional setiap unit bus di buku besar dan merekap data dari buku besar ke *excel* yang



dilakukan oleh bagian administrasi pembukuan. Proses bisnis yang ditambah adalah proses bisnis menampilkan laporan unit yang dilakukan oleh sistem.

#### 4.2.2 Tabel Perbandingan

Dalam Tabel 4.5 menjelaskan bahwa terdapat perbedaan waktu lama dari pelaksanaan proses bisnis yang terdapat pada PT Mahkota Mitra Sentosa. Waktu lama pada bisnis proses *as-is* didapatkan dari hasil wawancara pada bagian administrasi serta observasi langsung pada PT Mahkota Mitra Sentosa. Sedangkan untuk perkiraan waktu lama pada bisnis proses *to-be* yaitu peneliti melakukan perkiraan waktu yang berjalan ketika menggunakan bisnis proses yang diusulkan. Informasi pada Tabel 4.5 merupakan hasil dari simulasi waktu menggunakan perangkat bantuan yaitu bizagi.

**Tabel 4.5 Perbandingan Waktu Pelaksanaan Proses Bisnis**

No.	Proses Bisnis	<i>As-is</i>		<i>To-be</i>	
		Rata-rata waktu (menit)	Total waktu (menit)	Perkiraan rata-rata waktu (menit)	Perkiraan total waktu (menit)
1.	Sewa bus	8.8	186.6	8.8	152
2.	Membuat slip gaji	4.1	112.5	0.8	23.85
3.	Membuat laporan unit	29.5	796.5	0.7	19.35
4	Membuat laporan global	18.5	499.5	0.7	19.35

#### 4.3 Vision Document

*Vision document* menjelaskan analisis dari permasalahan yang dihadapi sebelum kebutuhan diambil. Pada *vision document*, elemen-elemen yang dianalisis adalah tipe pengguna dan pemangku kepentingan, pernyataan permasalahan, kebutuhan pemangku kepentingan, dan kedudukan produk. Sehingga dapat menghasilkan fitur sistem yang nantinya dikembangkan menjadi kebutuhan sistem.

##### 4.3.1 Identifikasi Tipe Pengguna dan Pemangku Kepentingan

Identifikasi tipe pemangku dan pemangku kepentingan mencakup tipe pengguna dan pemangku kepentingan, serta informasi pengguna sistem. Tipe pemangku kepentingan yang dimasukkan adalah pemangku kepentingan berkaitan dengan proses bisnis usulan.

#### 4.3.1.1 Tipe Pengguna dan Pemangku Kepentingan

Bagian ini menjelaskan tentang tipe-tipe pemangku kepentingan yang berkaitan dengan sistem informasi yang dibangun serta perwakilan-perwakilan pemangku kepentingan yang mewakili tiap tipe pemangku kepentingan.

**Tabel 4.6 Tabel Tipe Pengguna dan Pemangku Kepentingan**

No.	Tipe pemangku kepentingan	Deskripsi	Perwakilan pemangku kepentingan
1.	Pengguna	Orang yang memiliki peran langsung untuk menggunakan sistem. Peran pengguna akan secara langsung didefinisikan sebagai aktor pada <i>use-case</i>	Direktur, komisaris, bagian administrasi, pemasaran
2.	Pengembang	Orang yang memiliki andil dalam mengembangkan dan memproduksi sistem informasi penggajian karyawan	Peneliti

Tabel 4.6 merupakan Tabel tipe pemangku kepentingan beserta deskripsi dan perwakilan tiap tipe pemangku kepentingan. Pada Tabel tersebut, untuk tipe pengguna didapatkan empat perwakilan pemangku kepentingan, untuk tipe pengembang didapatkan satu perwakilan pemangku kepentingan. Dalam fase *elaboration* terdapat penambahan perwakilan pemangku kepentingan yaitu pemasaran.

#### 4.3.1.2 Informasi Pengguna

Bagian ini menjelaskan mengenai informasi yang berkaitan dengan pengguna sistem. Informasi tersebut adalah karakteristik tiap pengguna dan kompetensi yang dibutuhkan pengguna dalam menjalankan sistem.

**Tabel 4.7 Tabel Informasi Pengguna**

No.	Nama	Karakteristik	Kompetensi
1.	Direktur	Memiliki tanggung jawab manajemen seluruh kegiatan PT	Dapat mengoperasikan <i>browser</i> di HP maupun PC.

Tabel 4.7 Tabel Informasi Pengguna

No.	Nama	Karakteristik	Kompetensi
		Mahkota Mitra Sentosa	
2.	Komisaris	Memiliki tanggung jawab manajemen seluruh kegiatan PT Mahkota Mitra Sentosa. Sebagai wakil dari direktur.	Dapat mengoperasikan <i>browser</i> di HP maupun PC.
3.	Bagian administrasi (kasir)	Memiliki tanggung jawab manajemen persewaan dan gaji sopir kernet	Dapat mengoperasikan <i>browser</i> di HP maupun PC.
4.	Bagian administrasi (pembukuan)	Memiliki tanggung jawab manajemen pembukuan persewaan.	Dapat mengoperasikan <i>browser</i> di HP maupun PC.
5.	Bagian Pemasaran	Memiliki tanggung jawab memasarkan jasa PT Mahkota Mitra Sentosa kepada masyarakat	Dapat mengoperasikan <i>browser</i> di HP.

Tabel 4.7 merupakan Tabel yang berisikan informasi yang berkaitan dengan masing-masing dari pengguna yang didapatkan dari Tabel 4.6. Terdapat lima pengguna yang akan menggunakan sistem informasi dengan dengan karakteristik masing-masing dan kompetensi yang harus dimiliki pengguna. Pada fase *elaboration* terdapat penambahan pengguna yaitu bagian pemasaran.

#### 4.3.2 Identifikasi Perwakilan Pemangku Kepentingan

Bagian ini menjelaskan mengenai tanggung jawab dan keterlibatan dalam pengembangan sistem dari tiap perwakilan pemangku kepentingan yang didapatkan pada Tabel 4.7.

Tabel 4.8 Tabel Identifikasi Perwakilan Pemangku Kepentingan

No.	Nama	Deskripsi	Tanggung Jawab	Keterlibatan
1.	Direktur	Memiliki tanggung jawab manajemen seluruh kegiatan PT Mahkota Mitra Sentosa	Memantau jalannya pengembangan sistem. Memastikan bahwa sistem dengan mudah digunakan dan sesuai dengan kebutuhan direktur.	Penyedia informasi kebutuhan selaku direktur.
2.	Komisaris	Memiliki tanggung jawab manajemen seluruh kegiatan PT Mahkota Mitra Sentosa. Sebagai wakil dari direktur.	Memastikan bahwa sistem dengan mudah digunakan dan sesuai dengan kebutuhan komisaris.	Penyedia informasi kebutuhan selaku komisaris.
3.	Bagian Administrasi (pembukuan)	Memiliki tanggung jawab manajemen pembukuan persewaan.	Memastikan bahwa sistem dengan mudah digunakan dan sesuai dengan kebutuhan bagian administrasi (pembukuan).	Penyedia informasi kebutuhan selaku bagian administrasi (pembukuan).
4.	Bagian Administrasi (Kasir)	Memiliki tanggung jawab untuk melayani pelanggan, manajemen gaji sopir dan kernet.	Memastikan bahwa sistem dengan mudah digunakan dan sesuai dengan kebutuhan bagian	Penyedia informasi kebutuhan selaku bagian administrasi (kasir).

Tabel 4.8 Tabel Identifikasi Perwakilan Pemangku Kepentingan

No.	Nama	Deskripsi	Tanggung Jawab	Keterlibatan
			administrasi (kasir).	
5.	Bagian pemasaran	Memiliki tanggung jawab untuk melayani pelanggan dan memasarkan jasa kepada pelanggan.	Memastikan bahwa sistem dengan mudah digunakan dan sesuai dengan kebutuhan bagian pemasaran.	Penyedia informasi kebutuhan selaku bagian pemasaran.

Tabel 4.8 dijelaskan setiap pemangku kepentingan mempunyai tanggung jawab masing-masing. Dalam pengembangan sistem setiap pemangku mempunyai keterlibatan sebagai penyedia informasi sesuai dengan jabatannya. Dalam fase *elaboration* terdapat penambahan perwakilan pemangku kepentingan yaitu bagian pemasaran.

### 4.3.3 Pernyataan Masalah

Bagian ini menjelaskan mengenai masalah yang harus diselesaikan dengan adanya sistem. Pernyataan masalah dilakukan berdasarkan hasil wawancara yang direpresentasikan ke dalam notasi grafik BPMN. Pada tabel 4.9 telah dijelaskan bahwa terdapat permasalahan mengenai penyimpanan, pencarian data, dan perhitungan datanya masih menggunakan kalkulator. Sehingga dalam pengolahan data menjadi lama. Dari permasalahan tersebut memengaruhi pimpinan perusahaan.

Tabel 4.9 Tabel Pernyataan Masalah

Masalah	<ol style="list-style-type: none"> <li>1. Penyimpanan data persewaan masih berupa buku besar</li> <li>2. Pencarian data masih melalui buku besar</li> <li>3. Perhitungan data masih menggunakan kalkulator</li> <li>4. Penyimpanan data armada masih berupa buku</li> </ol>
Memengaruhi	<ol style="list-style-type: none"> <li>1. Direktur</li> <li>2. Komisaris</li> <li>3. Bagian administrasi</li> </ol>



Dampak	<ol style="list-style-type: none"> <li>1. Lama dalam pengelolaan data persewaan</li> <li>2. Lama dalam perhitungan gaji</li> <li>3. Lama dalam pembuatan laporan</li> </ol>
Solusi	<ol style="list-style-type: none"> <li>1. Memudahkan mengelola data persewaan</li> <li>2. Memudahkan perhitungan gaji</li> <li>3. Memudahkan pembuatan laporan</li> </ol>

#### 4.3.4 Kebutuhan Pemangku Kepentingan

Bagian ini merupakan pendefinisian kebutuhan tiap pemangku kepentingan yang didapatkan dari solusi pada Tabel 4.10. Pendefinisian kebutuhan yang dilakukan hanya bertujuan untuk menjelaskan mengapa kebutuhan sistem dibutuhkan oleh pemangku kepentingan.

**Tabel 4.10 Tabel Kebutuhan Pemangku Kepentingan**

No	Kebutuhan	Pemangku kepentingan	Prioritas	Situasi saat ini	Solusi yang diajukan
1	Kemudahan pengelolaan data persewaan	Bagian administrasi	Mo	Bagian administrasi dalam melakukan pengelolaan data persewaan secara manual yaitu melalui data pembukuan	Pengelolaan data persewaan dapat dilakukan melalui sistem secara <i>online</i>
2	Kemudahan pengelolaan data operasional	Bagian administrasi	Mo	Bagian administrasi dalam melakukan pengelolaan data operasional secara manual yaitu	Pengelolaan data operasional dapat dilakukan melalui sistem secara <i>online</i>

Tabel 4.10 Tabel Kebutuhan Pemangku Kepentingan

No	Kebutuhan	Pemangku kepentingan	Prioritas	Situasi saat ini	Solusi yang diajukan
				melalui data pembukuan	
3	Kemudahan pembuatan laporan	Bagian administrasi	Mo	Bagian administrasi dalam melakukan pembuatan laporan secara manual yaitu melalui data pembukuan kemudian direkap dalam <i>excel</i> .	Pembuatan laporan dapat dilakukan dengan menggunakan sistem. Jadi tidak perlu lagi melakukan perekapan di <i>excel</i>
4	Kemudahan perhitungan gaji	Bagian administrasi	Mo	Bagian administrasi dalam melakukan perhitungan gaji masih menggunakan kalkulasi manual.	Proses perhitungan gaji dilakukan secara otomatis
5	Kemudahan pengelolaan data armada	Bagian administrasi	Mo	Bagian administrasi dalam mengelola data armada masih secara manual yaitu	Pengelolaan data armada dapat dilakukan melalui sistem secara <i>online</i>

Tabel 4.10 Tabel Kebutuhan Pemangku Kepentingan

No	Kebutuhan	Pemangku kepentingan	Prioritas	Situasi saat ini	Solusi yang diajukan
				melalui data pembukuan	
6	Keamanan dan integrasi data	Bagian administrasi	S	-	Sistem dikembangkan dengan menyertakan proses autentifikasi
7	Sistem dapat diakses 24 jam sehari, 7 hari dalam seminggu	Bagian administrasi	Mo	-	Sistem dikembangkan secara <i>online</i>
8	Sistem dapat diakses pada PC dan smartphone dan pada bermacam-macam <i>browser</i>	Bagian administrasi	Mo	-	Sistem dikembangkan dengan mempertimbangkan dapat diakses oleh bermacam-macam <i>browser</i>

Tabel 4.10 merupakan kebutuhan-kebutuhan pemangku kepentingan. Pada bagian ini didapatkan delapan kebutuhan pemangku kepentingan dan didefinisikan pula mengapa alasan pernyataan tersebut dibutuhkan oleh masing-masing pemangku kepentingan pada kolom situasi saat ini dan solusi yang diajukan. Kemudian diberikan sebuah prioritas dari setiap kebutuhan pemangku kepentingan.

#### 4.3.5 Kedudukan Produk

Bagian ini menjelaskan kemampuan dan kedudukan produk secara umum jika dihadapkan dengan kebutuhan yang telah didefinisikan.

**Tabel 4.11 Tabel Kedudukan Produk**

Untuk	Pegawai PT Mahkota Mitra Sentosa
Yang	Melakukan pengelolaan data persewaan
Untuk	Pegawai PT Mahkota Mitra Sentosa
Yang	Melakukan pengelolaan data operasional
Untuk	Pegawai PT Mahkota Mitra Sentosa
Yang	Melakukan perhitungan gaji
Untuk	Pegawai PT Mahkota Mitra Sentosa
Yang	Melakukan pengelolaan data armada
Produk	Sistem yang akan dibangun bertujuan untuk mengotomasi proses-proses yang terjadi di PT Mahkota Mitra Sentosa
Yang	Menyediakan fitur untuk mempermudah proses pengelolaan persewaan, operasional, armada, dan perhitungan gaji
Tidak seperti	Pada saat ini, di mana proses tersebut dilakukan secara manual melalui pembukuan dan perhitungan masih menggunakan kalkulator
Produk kami	Tersedia selama 24 jam 7 hari dalam seminggu selama terhubung dengan internet dan pengelolaan data dilakukan melalui sistem.

Tabel 4.11 merupakan hasil kedudukan produk sistem informasi penggajian yang akan dibangun didapatkan dari kebutuhan dan solusi yang ditawarkan pada Tabel 4.10. Berdasarkan tabel tersebut produk dapat melakukan pengelolaan data persewaan, operasional, armada bus, dan perhitungan gaji. Produk yang dibangun akan dapat akses kapan pun dan dimanapun selama perangkat terhubung dengan jaringan internet.

#### 4.3.6 Fitur

Fitur menjelaskan tentang kesimpulan dari kemampuan sistem informasi yang akan dibangun. Dalam menjelaskan fitur, digunakan skala prioritas MoSCoW. Fitur didapatkan dari penjabaran kebutuhan pemangku kepentingan.

Tabel 4.12 Tabel Fitur

No	Kode Fitur	Deskripsi	Prioritas
1	FEAT1	Sistem menyediakan fasilitas bagi <i>user</i> untuk mengakses sistem sesuai hak akses.	Mo
2	FEAT2	Sistem menyediakan fasilitas untuk melihat jadwal armada yang telah disewa ataupun armada yang belum disewa	Mo
3	FEAT3	Sistem menyediakan fasilitas untuk mengelola data daftar harga tujuan seperti melihat, menghapus, menyunting data daftar harga tujuan mengenai tempat tujuan dan harga.	Mo
4	FEAT4	Sistem menyediakan fasilitas untuk mengelola data armada seperti melihat, menghapus, menyunting data armada mengenai nomor polisi, tahun, kapasitas kursi armada dan sopir-kernet.	Mo
5	FEAT5	Sistem menyediakan fasilitas untuk mengelola data persewaan seperti melihat, menghapus, menyunting mengenai nama penyewa, tanggal sewa, tujuan perjalanan, biaya operasional.	Mo
6	FEAT6	Sistem menyediakan fasilitas untuk mencetak laporan unit bus.	Mo
7	FEAT7	Sistem menyediakan fasilitas untuk mengelola laporan akhir global.	Mo
8	FEAT8	Sistem menyediakan fasilitas untuk mencetak slip gaji.	Mo
9	FEAT9	Sistem menyediakan fasilitas untuk mengelola data pegawai seperti melihat, menambah, menghapus dan menyunting.	Mo
10	FEAT10	Sistem menyediakan fasilitas untuk mengelola data pegawai yang dapat menggunakan sistem seperti	S



No	Kode Fitur	Deskripsi	Prioritas
		melihat, menambah, menghapus, dan menyunting.	
11	FEAT11	Sistem menyediakan fasilitas untuk menyunting profil pengguna sistem.	S
12	FEAT12	Sistem menyediakan fasilitas untuk dapat diakses dari berbagai <i>browser</i>	Mo
13	FEAT13	Sistem menyediakan fasilitas untuk dapat diakses 24 jam sehari dan 7 hari seminggu	Mo
14	FEAT14	Sistem menyediakan fasilitas keamanan data	Mo

Tabel 4.12 berisikan fitur-fitur sistem yang didapatkan dari analisis kebutuhan pengguna pada Tabel 4.10. Pada penelitian ini didapatkan 14 fitur pada sistem. Setiap fitur mempunyai kriteria prioritas masing-masing.

#### 4.4 Kebutuhan Fungsional dan Non Fungsional

Kebutuhan sistem dibagi menjadi dua jenis, yaitu kebutuhan fungsional dan kebutuhan non fungsional.

##### 4.4.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang harus tersedia untuk memenuhi kebutuhan pengguna yang telah didefinisikan pada Tabel 4.13. Tabel tersebut merupakan kebutuhan fungsional sistem yang dijadikan patokan dalam pembuatan sistem. Kebutuhan fungsional didapatkan dari pengembangan fitur pada Tabel 4.12. Setiap fitur yang telah didefinisikan mempunyai beberapa fungsionalitas masing-masing.

**Tabel 4.13 Tabel Kebutuhan Fungsional**

Kode Fitur	Kode Fungsi	Nama Fungsi	Deskripsi
FEAT01	SKPL-F-01	Log In	Sistem menyediakan fasilitas bagi <i>user</i> untuk masuk ke dalam sistem sesuai dengan hak akses.
	SKPL-F-02	Log Out	Sistem menyediakan fasilitas bagi <i>user</i> untuk keluar dari sistem.

Kode Fitur	Kode Fungsi	Nama Fungsi	Deskripsi
FEAT02	SKPL-F-03	Melihat informasi jadwal armada	Sistem dapat menampilkan informasi mengenai jadwal armada bus.
FEAT03	SKPL-F-04	Menambah armada bus	Sistem dapat menambah data armada bus mengenai nomor polisi, tahun, kapasitas kursi armada dan sopir-kernet.
	SKPL-F-05	Menghapus armada bus	Sistem dapat menghapus data armada bus mengenai nomor polisi, tahun, kapasitas kursi armada dan sopir-kernet.
	SKPL-F-06	Menyunting armada bus	Sistem dapat menyunting data armada bus mengenai nomor polisi, tahun, kapasitas kursi armada dan sopir-kernet.
FEAT04	SKPL-F-07	Menambah daftar tujuan	Sistem dapat menambah data daftar tujuan mengenai tempat tujuan dan harga.
	SKPL-F-08	Menghapus daftar tujuan	Sistem dapat menghapus data daftar tujuan mengenai tempat tujuan dan harga.
	SKPL-F-09	Menyunting daftar tujuan	Sistem dapat menyunting data daftar tujuan mengenai tempat tujuan dan harga.
FEAT05	SKPL-F-10	Menambah persewaan	Sistem dapat menambah data persewaan meliputi, nama penyewa, tanggal sewa, tanggal lunas, tanggal berangkat harga sewa, biaya DP, tujuan, jam berangkat, tempat penjemputan

Kode Fitur	Kode Fungsi	Nama Fungsi	Deskripsi
	SKPL-F-11	Menghapus persewaan	Sistem dapat menghapus data persewaan meliputi, nama penyewa, tanggal sewa, tanggal lunas, tanggal berangkat harga sewa, biaya DP, tujuan, jam berangkat, tempat penjemputan
	SKPL-F-12	Menyunting persewaan	Sistem dapat menyunting data persewaan meliputi, nama penyewa, tanggal sewa, tanggal lunas, tanggal berangkat harga sewa, biaya DP, tujuan, jam berangkat, tempat penjemputan
FEAT06	SKPL-F-13	Mencetak laporan unit bus	Sistem dapat melakukan fungsi cetak data laporan akhir bulan per unit bus.
FEAT07	SKPL-F-14	Mencetak laporan global bus	Sistem dapat melakukan fungsi cetak data laporan global bus.
FEAT08	SKPL-F-15	Mencetak slip gaji	Sistem dapat melakukan fungsi cetak rincian slip gaji sopir-kernet.
FEAT09	SKPL-F-16	Menambah pegawai	Sistem dapat menambah pegawai
	SKPL-F-17	Menghapus pegawai	Sistem dapat menghapus data pegawai
	SKPL-F-18	Menyunting pegawai	Sistem dapat menyunting data pegawai
FEAT10	SKPL-F-19	Menambah pengguna sistem	Sistem dapat menambah pengguna sistem
	SKPL-F-20	Menghapus pengguna sistem	Sistem dapat menghapus data pengguna sistem
	SKPL-F-21	Menyunting pengguna sistem	Sistem dapat menyunting data pengguna sistem
FEAT11	SKPL-F-22	Menyunting profil	Sistem dapat menyunting profil

#### 4.4.2 Kebutuhan Non Fungsional

Pada penelitian ini kebutuhan non fungsional tidak dibahas lebih lanjut karena lebih difokuskan pada kebutuhan fungsional.

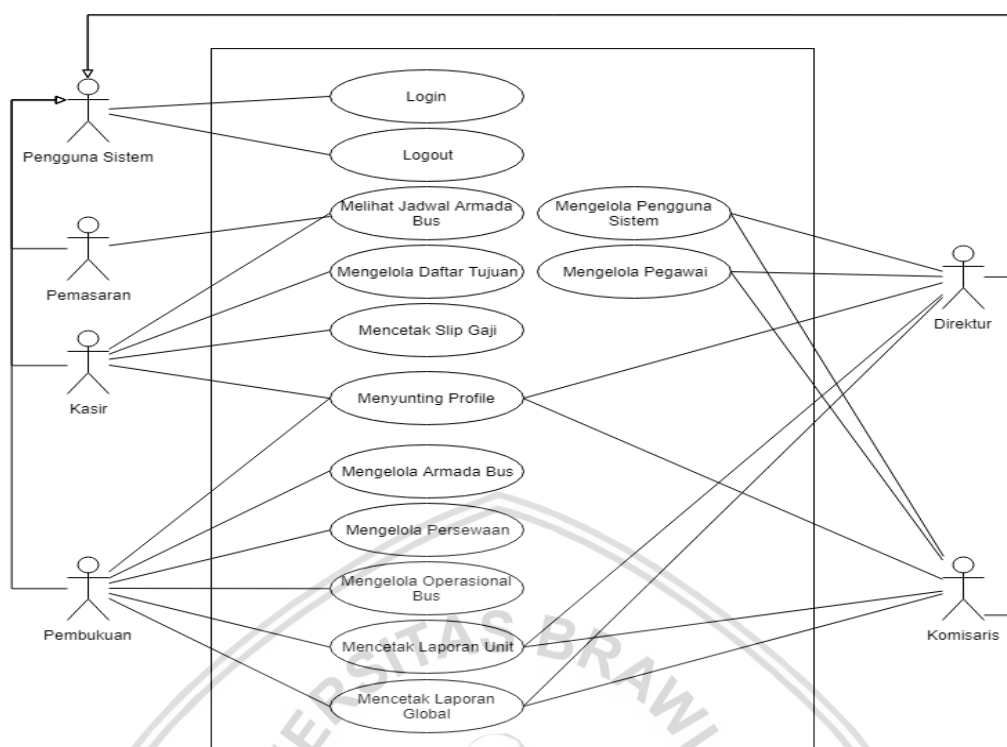
**Tabel 4.14 Tabel Kebutuhan Non Fungsional**

Kode Fitur	Kode Fungsi	Nama Fungsi	Deskripsi
FEAT12	SKPL-NF-01	<i>Compatibility</i>	Sistem dapat berjalan diberbagai <i>browser</i> yang berbeda.
FEAT13	SKPL-NF-02	Waktu akses	Sistem dapat diakses 24 jam sehari dan 7 hari seminggu
FEAT14	SKPL-NF-03	Keamanan data dan integrasi data	Keamanan menggunakan encoding md5

Tabel 4.14 merupakan kebutuhan non fungsional dari sistem yang akan dibuat yang didapatkan dari pengembangan fitur pada Tabel 4.12. Dalam kebutuhan non fungsional terdapat tiga dengan nama fungsi yaitu *compatibility*, waktu akses, keamanan data dan integrasi data. Setiap fungsi yang ada diberikan kode fungsi masing-masing.

#### 4.5 Pemodelan *Use-case*

Pada bagian ini dilakukan pemodelan *use-case*. Pemodelan *use-case* dilakukan dengan menggunakan *use-case diagram* dan menggambarkan bagaimana hubungan antara *use-case* dengan aktor.



**Gambar 4.9 Diagram Use-case**

Gambar 4.9 merupakan diagram *use-case* beserta aktor yang telah ditemukan pada penelitian ini. *Use-case* didapatkan dari kebutuhan fungsioinal pada Tabel 4.13 sedangkan aktor didapatkan dari pengguna sistem pada Tabel 4.7. Gambar tersebut merupakan *use-case diagram* setelah terjadi iterasi pada fase *elaboration*. Dalam fase tersebut terdapat penambahan aktor yaitu pemasaran.

#### 4.5.1 Deskripsi Aktor

Pada bagian ini dideskripsikan aktor-aktor yang dicantumkan pada diagram usecase.

**Tabel 4.15 Tabel Deskripsi Aktor**

No	Nama aktor	Deskripsi
1.	Direktur	Aktor direktur merupakan petinggi dalam struktur organisasi PT Mahkota Mitra Sentosa. Direktur mempunyai wewenang seperti <i>login</i> , <i>logout</i> , mengelola pegawai, mengelola pengguna sistem, mencetak laporan, dan menyunting profil.
2.	Komisaris	Aktor komisaris merupakan wakil dari direktur di mana mempunyai wewenang sama seperti direktur.



3.	Bagian administrasi (kasir)	Aktor kasir mempunyai wewenang dalam melakukan <i>login</i> , <i>logout</i> , melihat jadwal armada, mengelola daftar tujuan, mencetak slip gaji, dan menyunting profil.
4.	Bagian administrasi (pembukuan)	Aktor pembukuan mempunyai wewenang dalam melakukan <i>login</i> , <i>logout</i> , mengelola armada bus, mengelola persewaan, mengelola operasional bus, mencetak laporan, dan dan menyunting profil.
5.	Bagian pemasaran	Aktor pembukuan mempunyai wewenang dalam melakukan <i>login</i> , <i>logout</i> , melihat jadwal armada, dan menyunting profil.

Tabel 4.15 mendeskripsikan setiap aktor yang akan berinteraksi dengan sistem. Aktor-aktor tersebut merupakan perwakilan pengguna yang berada di PT Mahkota Mitra Sentosa. Pada fase *elaboration* terdapat penambahan aktor yaitu bagian pemasaran yang mempunyai akses ke sistem untuk melakukan *login*, *logout*, melihat jadwal armada, dan menyunting profil.

#### 4.5.2 Skenario Use-case

Pada bagian ini, penjelasan lebih lanjut mengenai setiap *use-case* yang telah didefinisikan pada diagram *use-case*.

##### 4.5.2.1 Use-case Melakukan Login

Penjelasan mengenai *scenario* dari kebutuhan fungsional *Login* akan dijelaskan pada Tabel 4.16 sebagai berikut:

**Tabel 4.16 Skenario Use-Case Melakukan Login**

Flow of Events untuk <i>use-case</i> melakukan <i>login</i>	
<i>Brief Description</i>	<i>Use-case</i> ini menggambarkan bagaimana aktor masuk ke dalam sistem
<i>Actor</i>	Direktur, komisaris, kasir, pembukuan, pemasaran
<i>Pre-conditions</i>	1. Pengguna sistem telah terdaftar dan memiliki akun dalam sistem 2. Halaman <i>login</i> telah diakses oleh aktor
<i>Basic flow of Events</i>	1. Sistem menampilkan halaman <i>login</i> <i>Use-case</i> ini dimulai ketika sistem menampilkan halaman <i>login</i> . 2. Aktor memasukkan <i>username</i> dan <i>password</i>

	<p>Setelah berada dalam halaman <i>login</i>. Aktor memasukkan data berupa <i>username</i> dan <i>password</i>.</p> <p>3. Aktor menjalankan fungsi <i>login</i> Pada tahap ini setelah memasukkan data, aktor menjalankan fungsi <i>login</i>.</p> <p>4. Sistem melakukan autentikasi Sistem akan melakukan autentikasi dengan data yang telah di masukkan oleh aktor.</p> <p>5. Menampilkan halaman <i>dashboard</i> Jika data <i>username</i> dan <i>password</i> telah sesuai dengan data sistem, maka sistem akan menampilkan halaman <i>dashboard</i></p>
<i>Alternative Flow</i>	<p><b>A1. Unidentified User</b> Pada langkah 4 dalam <i>basic flow</i>. Jika sistem tidak menemukan kecocokan data yang telah dimasukkan maka sistem akan menampilkan pesan gagal <i>login</i>. Kembali ke tahap 2 pada <i>basic flow</i>.</p>
<i>Scenario</i>	<p>1. <i>Login: Basic flow</i> 2. <i>Unidentified User: Basic flow, Unidentified User</i></p>
<i>Post-conditions</i>	Aktor masuk ke dalam sistem dan sistem menampilkan halaman utama.

Pada tabel 4.16 merupakan urutan alur dari proses melakukan *login* pada sistem. Aktor yang dapat menjalankan adalah aktor yang memiliki hak untuk melakukan *login* kepada sistem. Dalam melakukan *login* aktor harus memasukkan nama pengguna dan kata sandi. Ketika data yang dimasukkan benar maka sistem akan menampilkan halaman utama dari sistem. *Alternative flow* A1 akan dijalankan ketika sistem tidak menemukan kecocokan data yang telah dimasukkan dengan *database*.

#### 4.5.2.2 Skenario *Use-Case* Melakukan *Logout*

Penjelasan mengenai skenario dari kebutuhan fungsional *Logout* akan dijelaskan pada Tabel 4.17 sebagai berikut:

**Tabel 4.17 Skenario *Use-Case* Melakukan *Logout***

Flow of Events untuk <i>use-case</i> melakukan <i>logout</i>	
<i>Brief Description</i>	<i>Use-case</i> ini menggambarkan bagaimana aktor keluar dari sistem
<i>Actor</i>	Direktur, komisaris, kasir, pembukuan, pemasaran

<i>Pre-conditions</i>	Aktor telah teridentifikasi sebagai pengguna sistem
<i>Basic flow of Events</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih fungsi <i>logout</i> <i>Use-case</i> ini dimulai ketika aktor memilih fungsi <i>logout</i>.</li> <li>2. Sistem menampilkan halaman <i>login</i> Sistem akan menjalankan fungsi <i>logout</i> dan sistem akan menampilkan halaman <i>login</i>.</li> </ol>
<i>Alternative Flows</i>	-
<i>Scenario</i>	<i>Logout : Basic flow</i>
<i>Post-conditions</i>	Aktor keluar dari sistem dan sistem menampilkan halaman <i>login</i> .

Pada tabel 4.17 merupakan urutan alur dari proses melakukan *logout* atau keluar dari sistem. Proses ini dapat dilakukan ketika pengguna sistem telah teridentifikasi sebagai pengguna. Dalam melakukan *logout*, pengguna cukup dengan memilih fungsi *logout* pada sistem. Kemudian, sistem akan menampilkan halaman *login*.

#### 4.5.2.3 Skenario *Use-Case* Melihat Jadwal Armada Bus

Penjelasan mengenai *scenario* dari kebutuhan fungsional melihat jadwal armada bus akan dijelaskan pada Tabel 4.18 yaitu sebagai berikut

**Tabel 4.18 Skenario *Use-Case* Melihat Jadwal Armada Bus**

Flow of Events untuk <i>use-case</i> melihat jadwal armada bus	
<i>Brief Description</i>	<i>Use-case</i> ini menggambarkan bagaimana aktor melihat jadwal armada bus
<i>Actor</i>	Kasir, pemasaran
<i>Pre-conditions</i>	Aktor telah teridentifikasi sebagai pengguna sistem
<i>Basic flow of Events</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih fungsi jadwal armada bus. <i>Use-case</i> ini dimulai ketika aktor memilih fungsi jadwal armada bus. Sistem akan mengarahkan ke halaman jadwal armada bus.</li> <li>2. Sistem menampilkan halaman jadwal armada bus. Sistem menampilkan halaman armada armada bus yang berisi <i>form</i> masukkan tanggal.</li> <li>3. Aktor memasukkan tanggal Setelah sistem menampilkan <i>form</i> masukkan. Aktor memasukkan data tanggal yang ingin dicari</li> <li>4. Aktor memilih fungsi cari</li> </ol>

	Aktor memilih fungsi cari untuk mencari data yang sesuai masukkan data tanggal.  5. Sistem menampilkan data armada bus Setelah aktor memilih fungsi cari, sistem akan menampilkan data jadwal armada bus yang telah disewa maupun yang belum disewa sesuai dengan masukkan data tanggal.
<i>Alternative Flows</i>	-
<i>Scenario</i>	Melihat jadwal armada bus: <i>Basic flow</i>
<i>Post-conditions</i>	Sistem menampilkan data armada bus yang telah disewa maupun yang belum disewa sesuai dengan masukkan tanggal aktor dan mendapatkan informasi mengenai jadwal armada bus.

Pada tabel 4.18 merupakan urutan alur dari proses melihat jadwal bus. Proses ini dapat dilakukan ketika pengguna sistem telah teridentifikasi sebagai pengguna. Aktor akan memilih fungsi jadwal bus. Kemudian sistem akan menampilkan *form* masukkan dan aktor akan mengisi *form* tersebut. Sistem akan menampilkan data jadwal armada bus sesuai dengan masukkan dari aktor ke sistem setelah aktor memilih fungsi jadwal bus dan mendapatkan informasi mengenai jadwal armada bus.

#### 4.5.2.4 Skenario *Use-Case* Mengelola Armada Bus

Penjelasan mengenai *scenario* dari kebutuhan fungsional mengelola armada bus akan dijelaskan pada Tabel 4.19 sebagai berikut:

**Tabel 4.19 Skenario *Use-Case* Mengelola Armada Bus**

Flow of Events untuk <i>use-case</i> mengelola armada bus	
<i>Brief Description</i>	<i>Use-case</i> ini menggambarkan bagaimana aktor mengelola armada bus
<i>Actor</i>	Pembukuan
<i>Pre-conditions</i>	Aktor telah teridentifikasi sebagai pengguna sistem
<i>Basic flow of Events</i>	<ol style="list-style-type: none"> <li>1. Aktor sistem memilih fungsi armada bus. <i>Use-case</i> ini dimulai ketika aktor memilih fungsi armada bus. Sistem akan mengarahkan ke halaman armada bus</li> <li>2. Sistem menampilkan halaman armada bus Sistem akan mengambil rekam armada bus dari database. Kemudian, sistem akan menampilkan list daftar armada bus.</li> <li>3. Menambah armada bus</li> </ol>

	<p>Setelah sistem menampilkan list daftar armada bus. Sistem juga menampilkan beberapa fungsi yang tersedia seperti: Menambah, Menyunting dan Menghapus Armada Bus. Dalam hal ini aktor memilih fungsi tambah armada bus.</p> <p>4. Sistem menampilkan halaman tambah armada bus Saat aktor memilih fungsi menambah armada bus, sistem akan menampilkan halaman tambah armada bus. Halaman ini berisi <i>form</i> masukkan data armada.</p> <p>5. Mengisi <i>form</i> Setelah sistem menampilkan <i>form</i> armada bus. Pengguna sistem memasukkan data armada bus dalam <i>form</i> masukkan yang telah ditampilkan oleh sistem.</p> <p>6. <i>Submit</i> data armada bus Aktor memilih fungsi <i>submit</i> data. Kemudian sistem akan menyimpan data armada bus yang telah diinputkan aktor ke dalam database. <i>Use-case</i> selesai.</p>
Alternative Flows	<p><b>A1: Menghapus armada bus</b> Pada langkah “Menambah armada bus” dalam <i>basic flow</i>. Setelah sistem halaman armada bus dan di mana aktor telah mempunyai armada bus yang telah tersimpan. aktor memilih daftar rekap armada yang akan dihapus dan menggunakan fungsi hapus. Sistem akan meminta aktor untuk melakukan konfirmasi menghapus daftar. Aktor mengkonfirmasi penghapusan. Sistem akan melakukan penghapusan daftar dalam database dan sistem akan menampilkan list daftar armada bus ke aktor. <i>Use-case</i> berakhir.</p> <p><b>A2: Menyunting armada bus</b> Pada langkah “Menambah armada bus” dalam <i>basic flow</i>. Setelah sistem halaman armada bus dan di mana aktor telah mempunyai armada bus yang telah tersimpan. aktor dapat menggunakan sebagai titik awal untuk menggunakan fungsi sunting armada bus tersebut. <i>Use-case</i> kembali pada langkah mengisi <i>form</i> yang disediakan dalam <i>basic flow</i>.</p> <p><b>A3: Incomplete Form</b></p>

	Pada langkah “ <i>Submit data armada bus</i> ” dalam <i>basic flow</i> . Jika sistem menentukan bahwa identifikasi <i>form</i> armada bus yang dikirimkan tidak lengkap maka sebuah pesan berupa error message akan ditampilkan dan <i>use-case</i> kembali pada langkah mengisi <i>form</i> pada <i>basic flow</i> .
<i>Scenario</i>	<ol style="list-style-type: none"> <li>1. Menambah armada bus: <i>Basic flow</i></li> <li>2. Menghapus armada bus: <i>Basic flow</i>, Menghapus armada bus</li> <li>3. Menyunting armada bus: <i>basic flow</i>, Menyunting armada bus</li> </ol>
<i>Post-conditions</i>	Sistem menampilkan data armada

Pada tabel 4.19 merupakan urutan alur dari proses mengelola armada bus. Proses ini dapat dilakukan ketika pengguna sistem telah teridentifikasi sebagai pengguna. Untuk mengelola armada aktor memilih fungsi armada bus. Ketika sistem sudah menampilkan halaman armada bus, aktor dapat melakukan tambah armada, hapus armada, dan menyunting armada yang sudah ada dengan memilih fungsi yang telah tersedia. Ketika aktor akan melakukan hapus armada maka aktor menjalankan *alternative flows* A1. Aktor akan menjalankan *alternative flow* A2 ketika aktor akan menyunting armada bus. *Alternative flow* A3 akan dijalankan ketika *form* masukkan belum diisi secara benar.

#### 4.5.2.5 Skenario *Use-Case* Mengelola Daftar Tujuan

Penjelasan mengenai *scenario* dari kebutuhan fungsional mengelola daftar tujuan akan dijelaskan pada Tabel 4.20 sebagai berikut:

**Tabel 4.20 Skenario *Use-Case* Mengelola Daftar Tujuan**

Flow of Events untuk <i>use-case</i> mengelola daftar tujuan	
<i>Brief Description</i>	<i>Use-case</i> ini menggambarkan bagaimana aktor mengelola daftar tujuan
<i>Actor</i>	Pembukuan
<i>Pre-conditions</i>	Aktor telah teridentifikasi sebagai pengguna sistem
<i>Basic flow of Events</i>	<ol style="list-style-type: none"> <li>1. Aktor sistem memilih fungsi daftar tujuan. <i>Use-case</i> ini dimulai ketika aktor memilih fungsi daftar tujuan. Sistem akan mengarahkan ke halaman daftar tujuan.</li> <li>2. Sistem menampilkan halaman daftar tujuan Sistem akan mengambil rekam daftar tujuan dari database. Kemudian, sistem akan menampilkan list daftar daftar tujuan.</li> </ol>



	<p>3. Menambah daftar tujuan Setelah sistem menampilkan list daftar daftar tujuan. Sistem juga menampilkan beberapa fungsi yang tersedia seperti: Menambah, Menyunting dan Menghapus daftar tujuan. Dalam hal ini aktor memilih fungsi tambah daftar tujuan.</p> <p>4. Sistem menampilkan halaman tambah daftar tujuan Saat aktor memilih fungsi menambah daftar tujuan, sistem akan menampilkan halaman tambah daftar tujuan. Halaman ini berisi <i>form</i> masukkan data daftar tujuan.</p> <p>5. Mengisi <i>form</i> Setelah sistem menampilkan <i>form</i> daftar tujuan. Pengguna sistem memasukkan data daftar tujuan dalam <i>form</i> masukkan yang telah ditampilkan oleh sistem.</p> <p>6. <i>Submit</i> data daftar tujuan Aktor memilih fungsi <i>submit</i> data. Kemudian sistem akan menyimpan data daftar tujuan yang telah diinputkan aktor ke dalam database. <i>Use-case</i> selesai.</p>
Alternative Flows	<p><b>A1: Menghapus daftar tujuan</b> Pada langkah “Menambah daftar tujuan” dalam <i>basic flow</i>. Setelah sistem halaman daftar tujuan dan di mana aktor telah mempunyai daftar tujuan yang telah tersimpan. aktor memilih daftar rekap daftar tujuan yang akan dihapus dan menggunakan fungsi hapus. Sistem akan meminta aktor untuk melakukan konfirmasi menghapus daftar. Aktor mengkonfirmasi penghapusan. Sistem akan melakukan penghapusan daftar dalam database dan sistem akan menampilkan list daftar daftar tujuan ke aktor. <i>Use-case</i> berakhir.</p> <p><b>A2: Menyunting daftar tujuan</b> Pada langkah “Menambah daftar tujuan” dalam <i>basic flow</i>. Setelah sistem halaman daftar tujuan dan di mana aktor telah mempunyai daftar tujuan yang telah tersimpan. aktor dapat menggunakan sebagai titik awal untuk menggunakan fungsi sunting daftar tujuan</p>

	<p>tersebut. <i>Use-case</i> kembali pada langkah mengisi <i>form</i> yang disediakan dalam <i>basic flow</i>.</p> <p><b>A3: Incomplete Form</b>  Pada langkah “<i>Submit</i> data daftar tujuan” dalam <i>basic flow</i>. Jika sistem menentukan bahwa identifikasi <i>form</i> daftar tujuan yang dikirimkan tidak lengkap maka sebuah pesan berupa error message akan ditampilkan dan <i>use-case</i> kembali pada langkah mengisi <i>form</i> pada <i>basic flow</i>.</p>
<i>Scenario</i>	<ol style="list-style-type: none"> <li>1. Menambah daftar tujuan: <i>Basic flow</i></li> <li>2. Menghapus daftar tujuan: <i>Basic flow</i>, Menghapus daftar tujuan</li> <li>3. Menyunting daftar tujuan: <i>basic flow</i>, Menyunting daftar tujuan</li> </ol>
<i>Post-conditions</i>	Sistem menampilkan daftar tujuan

Pada tabel 4.20 merupakan urutan alur dari proses mengelola daftar tujuan. Proses ini dapat dilakukan ketika pengguna sistem telah teridentifikasi sebagai pengguna. Aktor memilih fungsi daftar tujuan pada sistem untuk mengelola daftar tujuan. Kemudian sistem akan menampilkan halaman daftar tujuan dan aktor dapat menambah daftar tujuan baru, menghapus, dan menyunting daftar tujuan yang telah ada dengan memilih fungsi yang telah tersedia. Ketika aktor akan melakukan hapus daftar tujuan maka aktor menjalankan *alternative flows* A1. Aktor akan menjalankan *alternative flow* A2 ketika aktor akan menyunting daftar tujuan. *Alternative flow* A3 akan dijalankan ketika *form* masukkan belum diisi secara benar.

#### 4.5.2.6 Skenario *Use-Case* Mengelola Persewaan

Penjelasan mengenai *scenario* dari kebutuhan fungsional mengelola persewaan akan dijelaskan pada Tabel 4.21 sebagai berikut:

**Tabel 4.21 Skenario *Use-Case* Mengelola Persewaan**

Flow of Events untuk <i>use-case</i> mengelola persewaan	
<i>Brief Description</i>	<i>Use-case</i> ini menggambarkan bagaimana aktor mengelola persewaan
<i>Actor</i>	Pembukuan
<i>Pre-conditions</i>	Aktor telah teridentifikasi sebagai pengguna sistem
<i>Basic flow of Events</i>	<ol style="list-style-type: none"> <li>1. Pembukuan memilih fungsi persewaan. Untuk mengelola persewaan, aktor memilih fungsi persewaan.</li> <li>2. Sistem menampilkan halaman persewaan</li> </ol>

	<p>Setelah aktor memilih fungsi persewaan, sistem akan menampilkan data persewaan yang telah terdaftar.</p> <p>3. Menambah persewaan Saat sistem menampilkan daftar persewaan. Sistem juga menampilkan beberapa fungsi yang tersedia seperti: menambah, menyunting dan menghapus persewaan. Dalam hal ini pembukuan memilih fungsi menambah persewaan.</p> <p>4. Sistem menampilkan halaman tambah persewaan Saat pembukuan memilih fungsi menambah persewaan, sistem akan menampilkan halaman tambah persewaan. Halaman ini berisi <i>form</i> masukkan data persewaan.</p> <p>5. Mengisi <i>form</i> Setelah sistem menampilkan <i>form</i> persewaan. Pada tahap ini aktor memasukkan data persewaan dalam <i>form</i> masukkan yang telah ditampilkan oleh sistem.</p> <p>6. <i>Submit</i> data persewaan Aktor memilih fungsi simpan data. Kemudian sistem akan menyimpan data persewaan yang telah diinputkan ke dalam database. <i>Use-case</i> selesai</p>
Alternative Flows	<p><b>A1: Menghapus persewaan</b> Pada langkah “Menambah persewaan” dalam <i>basic flow</i>. Setelah sistem halaman persewaan dan aktor telah mempunyai daftar persewaan yang telah tersimpan. Aktor memilih daftar rekap persewaan yang akan dihapus. Sistem akan meminta pengguna sistem untuk melakukan konfirmasi menghapus daftar. Aktor mengkonfirmasi penghapusan. Sistem akan melakukan penghapusan daftar dalam database. Lalu <i>use-case</i> berlanjut pada dan sistem akan menampilkan list daftar persewaan ke aktor. <i>Use-case</i> berakhir.</p> <p><b>A2: Menyunting persewaan</b> Pada langkah “Menambah persewaan” dalam <i>basic flow</i>. Setelah sistem menampilkan halaman persewaan dan di mana aktor telah mempunyai persewaan yang telah tersimpan. Aktor dapat menggunakan sebagai titik awal untuk menyunting armada bus tersebut. <i>Use-case</i></p>

	<p>kembali pada langkah mengisi <i>form</i> yang disediakan dalam <i>basic flow</i>.</p> <p><b>A3: Incomplete Form</b>  Pada langkah “<i>Submit data armada bus</i>” dalam <i>basic flow</i>. Jika sistem menentukan bahwa identifikasi <i>form</i> armada bus yang dikirimkan tidak lengkap maka sebuah pesan berupa error message akan ditampilkan dan <i>use-case</i> berakhir.</p>
<i>Scenario</i>	<ol style="list-style-type: none"> <li>1. Menambah persewaan: <i>Basic flow</i></li> <li>2. Menghapus persewaan: <i>Basic flow</i>, Menghapus persewaan</li> <li>3. Menyunting persewaan: <i>basic flow</i>, Menyunting persewaan</li> </ol>
<i>Post-conditions</i>	Sistem menampilkan data persewaan

Pada tabel 4.21 merupakan urutan alur dari proses mengelola persewaan. Proses ini dapat dilakukan ketika pengguna sistem telah teridentifikasi sebagai pengguna. Aktor memilih fungsi persewaan untuk dapat mengelola persewaan. Kemudian sistem akan menampilkan halaman persewaan. Aktor dapat menambah data persewaan baru, menghapus, dan menyunting data persewaan yang telah ada. Ketika aktor akan melakukan hapus persewaan maka aktor menjalankan *alternative flows* A1. Aktor akan menjalankan *alternative flow* A2 ketika aktor akan menyunting persewaan. *Alternative flow* A3 akan dijalankan ketika *form* masukkan belum diisi secara benar.

#### 4.5.2.7 Use-Case Skenario Mencetak Laporan Akhir Bulan per Unit Bus

Penjelasan mengenai *scenario* dari kebutuhan fungsional mencetak laporan akhir bulan per unit bus akan dijelaskan pada Tabel 4.22 sebagai berikut:

**Tabel 4.22 Use-Case Skenario Mencetak Laporan Akhir Bulan per Unit Bus**

Flow of Events untuk <i>use-case</i> mencetak laporan akhir bulan per unit bus	
<i>Brief Description</i>	<i>Use-case</i> ini menggambarkan bagaimana aktor mencetak laporan akhir bulan per unit bus
<i>Actor</i>	Direktur, komisaris, pembukuan
<i>Pre-conditions</i>	Aktor telah teridentifikasi sebagai pengguna sistem
<i>Basic flow of Events</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih fungsi laporan per unit bus. Untuk melakukan cetak laporan per unit bus, pengguna sistem memilih fungsi laporan per unit bus.</li> <li>2. Sistem menampilkan halaman laporan per unit bus</li> </ol>

	<p>Setelah aktor memilih fungsi laporan per unit bus, sistem akan menampilkan halaman laporan per unit bus. Halaman ini berisi <i>form</i> masukkan periode laporan dan unit bus.</p> <p>3. Aktor memasukkan data Aktor diminta untuk memasukkan data berupa periode laporan dan unit bus.</p> <p>4. Aktor memilih fungsi cari Aktor memilih fungsi cari untuk mendapatkan data laporan per unit bus sesuai dengan data yang telah dimasukkan.</p> <p>5. Sistem menampilkan laporan Setelah aktor melakukan fungsi <i>submit</i>, sistem akan menampilkan laporan per unit bus sesuai dengan periode dan unit bus yang telah dimasukkan.</p> <p>6. Aktor memilih fungsi cetak aktor dapat mencetak laporan per unit bus dengan menjalankan fungsi cetak. Sistem akan menampilkan halaman cetak</p> <p>7. Aktor mencetak aktor melakukan cetak dengan printer. <i>Use-case</i> selesai.</p>
<i>Alternative Flows</i>	-
<i>Scenario</i>	<p>1. Mencetak laporan per unit bus: <i>Basic flow</i></p> <p>2. Mengunduh dokumen <i>excel</i>: <i>Basic flow</i>, Mengunduh dokumen <i>excel</i></p> <p>3. Mengunduh dokumen <i>pdf</i>: <i>Basic flow</i>, Mengunduh dokumen <i>pdf</i></p>
<i>Post-conditions</i>	<p>1. Sistem menampilkan laporan akhir bulan per unit bus</p> <p>2. Sistem mencetak laporan akhir bulan per unit bus</p>

Pada tabel 4.22 merupakan urutan alur dari proses mencetak laporan akhir bulan per unit bus. Proses ini dapat dilakukan ketika pengguna sistem telah teridentifikasi sebagai pengguna. Aktor memilih fungsi laporan unit untuk dapat mencetak laporan. Kemudian sistem akan menampilkan halaman laporan unit dan aktor mengisi periode beserta nomor polisi pada *form* yang telah disediakan. Aktor dapat melakukan cetak laporan dengan memilih fungsi print dan aktor juga

dapat mengunduh laporan akhir menjadi *format pdf* ataupun dengan *format excel* dengan memilih fungsi *pdf* atau *excel*.

#### 4.5.2.8 Use-Case Skenario Mencetak Laporan Akhir Bulan Global

Penjelasan mengenai *scenario* dari kebutuhan fungsional mengelola laporan akhir bulan global akan dijelaskan pada Tabel 4.23 sebagai berikut:

**Tabel 4.23 Skenario Use-Case Mencetak Laporan Akhir Bulan Global**

Flow of Events untuk <i>use-case</i> mencetak laporan akhir bulan global	
<i>Brief Description</i>	<i>Use-case</i> ini menggambarkan bagaimana aktor mencetak laporan akhir bulan global
<i>Actor</i>	Direktur, komisaris, pembukuan
<i>Pre-conditions</i>	Aktor telah teridentifikasi sebagai pengguna sistem
<i>Basic flow of Events</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih fungsi laporan global. Untuk melakukan cetak laporan global, pengguna sistem memilih fungsi laporan global.</li> <li>2. Sistem menampilkan halaman laporan global Setelah aktor memilih fungsi laporan global, sistem akan menampilkan halaman laporan global. Halaman ini berisi <i>form</i> masukkan periode laporan global.</li> <li>3. Aktor memasukkan data Aktor diminta untuk memasukkan data berupa periode laporan.</li> <li>4. <i>Submit</i> periode Aktor memilih fungsi <i>submit</i> untuk mendapatkan data laporan global sesuai dengan data yang telah dimasukkan.</li> <li>5. Sistem menampilkan laporan Setelah aktor melakukan fungsi <i>submit</i>, sistem akan menampilkan laporan global sesuai dengan periode yang telah dimasukkan.</li> <li>6. Aktor memilih fungsi cetak Aktor dapat mencetak laporan global dengan menjalankan fungsi cetak. Sistem akan melakukan perintah cetak. <i>Use-case</i> selesai</li> </ol>
<i>Alternative Flows</i>	-
<i>Scenario</i>	<ol style="list-style-type: none"> <li>1. Mencetak laporan per unit bus: <i>Basic flow</i></li> <li>2. Mengunduh dokumen <i>excel</i>: <i>Basic flow</i>, Mengunduh dokumen <i>excel</i></li> </ol>



	3. Mengunduh dokumen <i>pdf</i> : <i>Basic flow</i> , Mengunduh dokumen <i>pdf</i>
<i>Post-conditions</i>	1. Sistem menampilkan laporan akhir bulan global 2. Sistem mencetak laporan akhir bulan global

Pada tabel 4.23 merupakan urutan alur dari proses mencetak laporan akhir bulan global. Proses ini dapat dilakukan ketika pengguna sistem telah teridentifikasi sebagai pengguna. Aktor memilih fungsi laporan global untuk dapat mencetak laporan. Kemudian sistem akan menampilkan halaman laporan global dan aktor mengisi periode pada *form* yang telah disediakan. Aktor dapat melakukan cetak laporan dengan memilih fungsi print dan aktor juga dapat mengunduh laporan akhir menjadi *format pdf* ataupun dengan *format excel* dengan memilih fungsi *pdf* atau *excel*.

#### 4.5.2.9 Skenario *Use-Case* Mencetak Slip Gaji

Penjelasan mengenai *scenario* dari kebutuhan fungsional mencetak slip gaji akan dijelaskan pada Tabel 4.24 sebagai berikut:

**Tabel 4.24 Skenario *Use-Case* Mencetak Slip Gaji**

Flow of Events untuk <i>use-case</i> mencetak slip gaji	
<i>Brief Description</i>	<i>Use-case</i> ini menggambarkan bagaimana aktor mencetak slip gaji
<i>Actor</i>	Kasir
<i>Pre-conditions</i>	Aktor telah teridentifikasi sebagai pengguna sistem
<i>Basic flow of Events</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih fungsi gaji Untuk melakukan cetak slip gaji, pengguna sistem memilih fungsi gaji.</li> <li>2. Sistem menampilkan halaman gaji Setelah aktor memilih fungsi gaji, sistem akan menampilkan halaman gaji. Halaman gaji berisi <i>form</i> masukkan kuitansi persewaan.</li> <li>3. Aktor memasukkan data Pengguna sistem diminta memasukkan data berupa nomor kuitansi persewaan sesuai dengan data persewaan dan memilih jenis armada bus.</li> <li>4. <i>Submit</i> data Aktor memilih fungsi <i>submit</i> untuk mendapatkan data gaji sesuai dengan data yang telah dimasukkan.</li> </ol>

	<p>5. Sistem menampilkan slip gaji Setelah aktor melakukan fungsi <i>submit</i>, sistem akan muncul alert konfirmasi dan menampilkan halaman slip gaji dengan detail rincian gaji sesuai dengan nomor kuitansi yang telah dimasukkan.</p> <p>6. Aktor memilih fungsi cetak Aktor dapat mencetak slip gaji dengan menjalankan fungsi cetak. Sistem akan menampilkan halaman cetak dan menampilkan <i>alert</i> konfirmasi.</p> <p>7. Aktor mencetak Aktor melakukan cetak dengan printer. <i>Use-case</i> selesai.</p>
<i>Alternative Flows</i>	<p><b>A1: Data tidak ditemukan</b> Pada langkah “<i>Submit data</i>” dalam <i>basic flow</i>. Jika sistem tidak menemukan data maka muncul pesan data tidak ditemukan dan <i>use-case</i> berakhir.</p>
<i>Scenario</i>	Mencetak slip gaji: <i>Basic flow</i>
<i>Post-conditions</i>	<p>1. Sistem menampilkan rincian gaji</p> <p>2. Sistem mencetak slip gaji</p>

Pada tabel 4.24 merupakan urutan alur dari proses mencetak slip gaji. Proses ini dapat dilakukan ketika aktor telah teridentifikasi sebagai pengguna sistem. Aktor memilih fungsi gaji untuk melakukan cetak. Kemudian sistem akan menampilkan halaman gaji dan aktor mengisi data pada *form* yang telah disediakan. Ketika aktor melakukan fungsi *submit*, sistem akan muncul *alert* konfirmasi dan menampilkan halaman slip gaji. Aktor dapat melakukan cetak slip gaji dengan memilih fungsi *print* yang terdapat pada halaman.

#### 4.5.2.10 Skenario *Use-Case* Mengelola Pegawai

Penjelasan mengenai *scenario* dari kebutuhan fungsional mengelola pegawai akan dijelaskan pada Tabel 4.25 sebagai berikut:

**Tabel 4.25 *Use-Case* Scenario Mengelola Pegawai**

Flow of Events untuk <i>use-case</i> mengelola pegawai	
<i>Brief Description</i>	<i>Use-case</i> ini menggambarkan bagaimana aktor mengelola pegawai
<i>Actor</i>	Direktur, komisaris
<i>Pre-conditions</i>	Aktor telah teridentifikasi sebagai pengguna sistem

Basic flow of Events	<ol style="list-style-type: none"> <li>1. Direktur/komisaris memilih fungsi pegawai Untuk mengelola pegawai, direktur/komisaris memilih fungsi pegawai.</li> <li>2. Sistem menampilkan halaman pegawai Setelah memilih fungsi pegawai, sistem akan menampilkan halaman pegawai. Halaman pegawai berisi daftar pegawai.</li> <li>3. Menambah pegawai Saat sistem menampilkan daftar pegawai. Sistem juga menampilkan beberapa fungsi yang tersedia seperti: menambah, menyunting dan menghapus pegawai. Dalam hal ini direktur/komisaris memilih fungsi tambah pegawai.</li> <li>4. Sistem menampilkan halaman tambah pegawai Setelah direktur/komisaris memilih fungsi menambah pegawai, sistem akan menampilkan halaman tambah pegawai. Halaman ini berisi <i>form</i> masukkan data pegawai.</li> <li>5. Mengisi <i>form</i> Pada tahap ini direktur/komisaris memasukkan data pegawai dalam <i>form</i> masukkan yang telah ditampilkan oleh sistem.</li> <li>6. <i>Submit</i> data pegawai Direktur/komisaris memilih fungsi <i>submit</i>. Kemudian sistem akan menyimpan data pegawai yang telah diinputkan direktur/komisaris ke dalam database. <i>Use-case</i> selesai.</li> </ol>
Alternative Flows	<p><b>A1: Menghapus pegawai</b> Pada langkah “Menambah pegawai” dalam <i>basic flow</i>. Setelah sistem menampilkan halaman pegawai dan di mana direktur/komisaris telah mempunyai daftar pegawai yang telah tersimpan. Direktur/komisaris memilih daftar rekap pegawai yang akan dihapus dan memilih fungsi hapus. Sistem akan meminta direktur/komisaris untuk melakukan konfirmasi menghapus daftar. Direktur/komisaris mengkonfirmasi penghapusan. Sistem akan melakukan penghapusan daftar dalam database dan sistem akan menampilkan daftar pegawai ke direktur/komisaris. <i>Use-case</i> berakhir.</p> <p><b>A2: Menyunting pegawai</b></p>

	<p>Pada langkah “Menambah pegawai” dalam <i>basic flow</i>. Setelah menampilkan sistem halaman pegawai dan di mana direktur/komisaris telah mempunyai daftar pegawai yang telah tersimpan. Direktur/komisaris dapat menggunakan sebagai titik awal untuk menyunting pegawai tersebut. <i>Use-case</i> kembali pada langkah mengisi <i>form</i> yang disediakan dalam <i>basic flow</i>.</p> <p><b>A3: Incomplete Form</b></p> <p>Pada langkah “Submit data pegawai” dalam <i>basic flow</i>. Jika sistem menentukan bahwa identifikasi <i>form</i> pegawai yang dikirimkan tidak lengkap maka sebuah pesan berupa error message akan ditampilkan dan <i>use-case</i> berakhir.</p>
Scenario	<ol style="list-style-type: none"> <li>1. Menambah pegawai: <i>Basic flow</i></li> <li>2. Menghapus pegawai: <i>Basic flow</i>, Menghapus pegawai</li> <li>3. Menyunting pegawai: <i>basic flow</i>, Menyunting pegawai</li> </ol>
Post-conditions	Sistem menampilkan data pegawai

Pada tabel 4.25 merupakan urutan alur dari proses mengelola pegawai. Proses ini dapat dilakukan ketika aktor telah teridentifikasi sebagai pengguna sistem. Aktor memilih fungsi pegawai untuk dapat mengelola pegawai. Kemudian sistem akan menampilkan halaman pegawai. Aktor dapat menambah data pegawai baru, menghapus, dan menyunting data pegawai yang telah ada. Ketika aktor akan melakukan hapus pegawai maka aktor menjalankan *alternative flows* A1. Aktor akan menjalankan *alternative flow* A2 ketika aktor akan menyunting pegawai. *Alternative flow* A3 akan dijalankan ketika *form* masukkan belum diisi secara benar.

#### 4.5.2.11 Use-Case Scenario Mengelola Pengguna Sistem

Penjelasan mengenai *scenario* dari kebutuhan fungsional mengelola pengguna sistem akan dijelaskan pada Tabel 4.26 sebagai berikut:

**Tabel 4.26 Use-Case Scenario Mengelola Pengguna Sistem**

Flow of Events untuk <i>use-case</i> mengelola pengguna sistem	
Brief Description	<i>Use-case</i> ini menggambarkan bagaimana aktor mengelola pengguna sistem

<i>Actor</i>	Direktur, komisaris
<i>Pre-conditions</i>	Aktor telah teridentifikasi sebagai pengguna sistem
<i>Basic flow of Events</i>	<ol style="list-style-type: none"> <li>1. Direktur/komisaris memilih fungsi pengguna sistem Untuk mengelola pegawai, direktur/komisaris memilih fungsi pengguna sistem.</li> <li>2. Sistem menampilkan halaman pengguna sistem Setelah memilih fungsi pengguna sistem, sistem akan menampilkan halaman pengguna sistem. Halaman pegawai berisi daftar pengguna sistem.</li> <li>3. Menambah pengguna sistem Saat sistem menampilkan daftar pengguna sistem. Sistem juga menampilkan beberapa fungsi yang tersedia seperti: menambah, menyunting dan menghapus pengguna sistem. Dalam hal ini direktur/komisaris memilih fungsi menambah pengguna sistem.</li> <li>4. Sistem menampilkan halaman tambah pengguna sistem Setelah direktur/komisaris memilih fungsi menambah pengguna sistem, sistem akan menampilkan halaman tambah pengguna sistem. Halaman ini berisi <i>form</i> masukkan data pengguna sistem.</li> <li>5. Mengisi <i>form</i> Pada tahap ini direktur/komisaris memasukkan data pengguna sistem dalam <i>form</i> masukkan yang telah ditampilkan oleh sistem.</li> <li>6. <i>Submit</i> data pengguna sistem Direktur/komisaris memilih fungsi <i>submit</i>. Kemudian sistem akan menyimpan data pengguna sistem yang telah diinputkan direktur/komisaris ke dalam database. <i>Use-case</i> selesai.</li> </ol>
<i>Alternative Flows</i>	<p><b>A1: Menghapus pengguna sistem</b> Pada langkah "Menambah pengguna sistem" dalam <i>basic flow</i>. Setelah sistem menampilkan halaman pengguna sistem dan di mana direktur/komisaris telah mempunyai daftar pegawai yang telah tersimpan. Direktur/komisaris memilih daftar rekap pengguna sistem yang akan dihapus. Sistem akan</p>

	<p>meminta direktur/komisaris untuk melakukan konfirmasi menghapus daftar. Direktur/komisaris mengkonfirmasi penghapusan. Sistem akan melakukan penghapusan daftar dalam database dan sistem akan menampilkan daftar pengguna sistem ke direktur/komisaris. <i>Use-case</i> berakhir.</p> <p><b>A2: Menyunting pengguna sistem</b>          Pada langkah “Menambah pengguna sistem” dalam <i>basic flow</i>. Setelah menampilkan sistem halaman pengguna sistem dan di mana direktur/komisaris telah mempunyai daftar pengguna sistem yang telah tersimpan. Direktur/komisaris dapat menggunakan sebagai titik awal untuk menyunting pengguna sistem tersebut. <i>Use-case</i> kembali pada langkah mengisi <i>form</i> yang disediakan dalam <i>basic flow</i>.</p> <p><b>A3: Incomplete Form</b>          Pada langkah “Submit data pengguna sistem” dalam <i>basic flow</i>. Jika sistem menentukan bahwa identifikasi <i>form</i> pengguna sistem yang dikirimkan tidak lengkap maka sebuah pesan berupa error message akan ditampilkan dan <i>use-case</i> berakhir.</p>
Scenario	<ol style="list-style-type: none"> <li>1. Menambah pengguna sistem: <i>Basic flow</i></li> <li>2. Menghapus pengguna sistem: <i>Basic flow</i>, Menghapus pengguna sistem</li> <li>3. Menyunting pengguna sistem: <i>basic flow</i>, Menyunting pengguna sistem</li> </ol>
Post-conditions	Sistem menampilkan data pengguna sistem

Pada tabel 4.26 merupakan urutan alur dari proses mengelola pengguna sistem. Proses ini dapat dilakukan ketika aktor telah teridentifikasi sebagai pengguna sistem. Aktor memilih fungsi pengguna sistem untuk dapat mengelola pengguna sistem. Kemudian sistem akan menampilkan halaman pengguna sistem. Aktor dapat menambah data pengguna sistem baru, menghapus, dan menyunting data pengguna sistem yang telah ada. Ketika aktor akan melakukan hapus pengguna sistem maka aktor menjalankan *alternative flows* A1. Aktor akan menjalankan *alternative flow* A2 ketika aktor akan menyunting pengguna sistem. *Alternative flow* A3 akan dijalankan ketika *form* masukkan belum diisi secara benar.



#### 4.5.2.12 Use-Case Scenario Menyunting Profil

Penjelasan mengenai *scenario* dari kebutuhan fungsional menyunting profil akan dijelaskan pada Tabel 4.27 yaitu sebagai berikut

**Tabel 4.27 Use-Case Scenario Menyunting Profil**

Flow of Events untuk <i>use-case</i> menyunting profil	
<i>Brief Description</i>	<i>Use-case</i> ini menggambarkan bagaimana aktor menyunting profil
<i>Actor</i>	Direktur, komisaris, kasir, pembukuan, pemasaran
<i>Pre-conditions</i>	Aktor telah teridentifikasi sebagai pengguna sistem
<i>Basic flow of Events</i>	<ol style="list-style-type: none"> <li>1. Pengguna sistem memilih fungsi sunting profil Untuk menyunting profil, pengguna sistem memilih fungsi sunting profil.</li> <li>2. Sistem menampilkan halaman sunting profil Setelah memilih fungsi sunting profil, sistem akan menampilkan halaman sunting profil. Halaman ini berisi <i>form</i> detail profil pengguna sistem.</li> <li>3. Mengisi <i>form</i> Pengguna sistem dapat menyunting profil dengan melakukan pengisian data profil pada <i>form</i> yang ditampilkan sistem.</li> <li>4. Simpan data profil Pengguna sistem memilih fungsi simpan. Kemudian sistem akan menyimpan data pembaharuan profil yang telah diinputkan pengguna sistem ke dalam database. <i>Use-case</i> selesai.</li> </ol>
<i>Alternative Flows</i>	<p><b>A1: Incomplete Form</b></p> <p>Pada langkah “Simpan data profil” dalam <i>basic flow</i>. Jika sistem menentukan bahwa identifikasi <i>form</i> profil yang dikirimkan tidak lengkap maka sebuah pesan berupa error message akan ditampilkan dan <i>use-case</i> berakhir.</p>
<i>Scenario</i>	Menyunting profil: <i>Basic flow</i>
<i>Post-conditions</i>	Sistem menampilkan data pengguna sistem

Pada tabel 4.27 merupakan urutan alur dari proses menyunting profil. Proses ini dapat dilakukan ketika aktor telah teridentifikasi sebagai pengguna sistem. Aktor memilih fungsi *my profile* untuk dapat menyunting profil. Kemudian sistem akan menampilkan halaman sunting profil. *Alternative flow* A1 akan

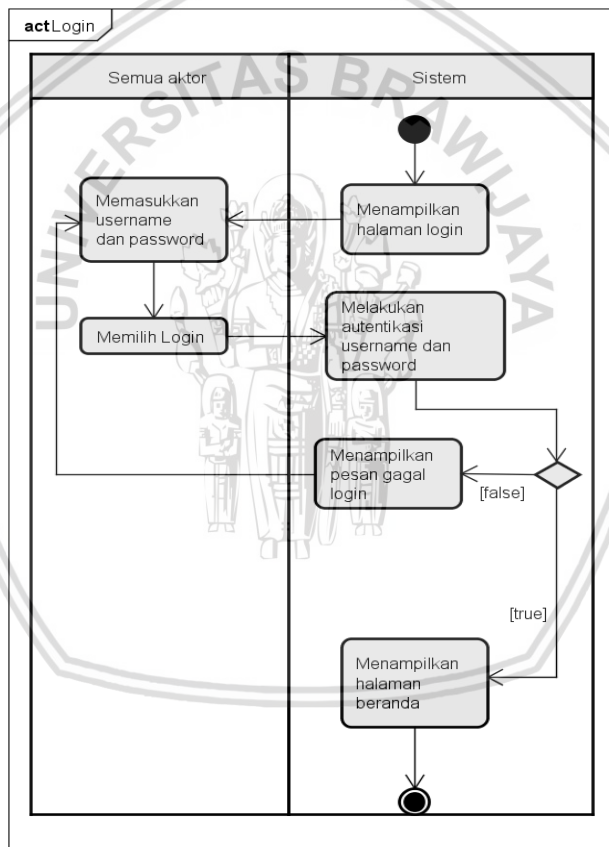
dijalankan ketika *form* masukkan belum diisi secara benar pada saat dilakukan simpan data.

## 4.6 Pemodelan Aktivitas

Diagram aktivitas merupakan salah satu jenis diagram yang digunakan untuk memodelkan alur proses yang didapatkan dari skenario *use-case*. *Diagram activity* menggambarkan hubungan antar elemen yang terjadi pada saat berjalannya suatu proses dari sudut pandang proses bisnis.

### 4.6.1 Activity Diagram Login

Aktivitas yang berjalan pada *use-case login* dapat dilihat pada Gambar 4.10 Aktivitas tersebut menggambarkan bagaimana proses *login* yang mengikuti proses pada *use-case scenario*.

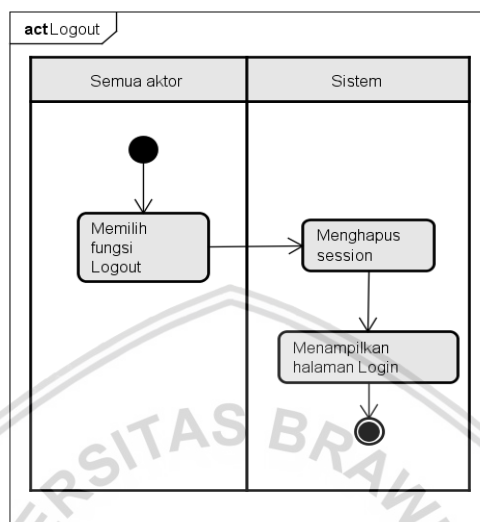


Gambar 4.10 Activity Diagram Login

Gambar 4.10 merupakan gambar *activity diagram login*. Aktivitas dimulai ketika sistem menampilkan halaman *login*. Kemudian aktor memasukkan *username, password* dan memilih fungsi *login*. Sistem akan memeriksa *username* dan *password* ketika data tersebut sesuai dengan yang ada pada sistem maka sistem akan menampilkan halaman beranda. Namun ketika data yang dimasukkan tidak sesuai dengan yang ada di sistem maka akan menampilkan pesan bahwa terjadi gagal *login*.

#### 4.6.2 Activity Diagram Logout

Aktivitas yang berjalan pada *use-case logout* dapat dilihat pada Gambar 4.11. Aktivitas tersebut menggambarkan bagaimana proses *logout* yang mengikuti proses pada *use-case scenario*

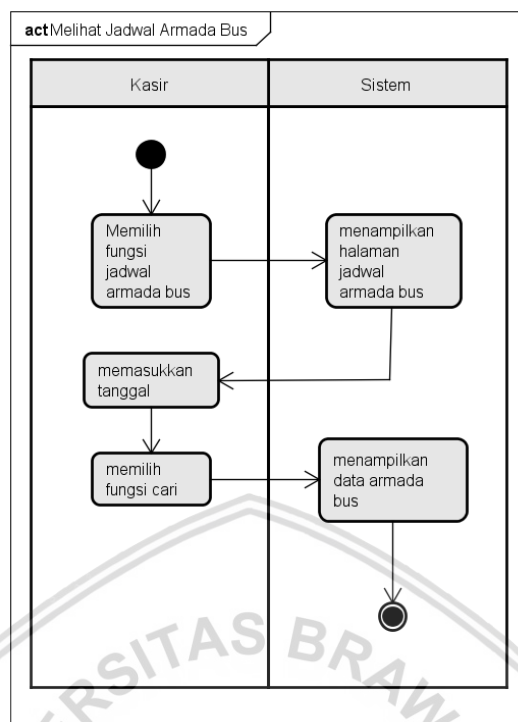


**Gambar 4.11 Activity Diagram Logout**

Gambar 4.11 merupakan gambar *activity diagram logout*. Aktivitas dimulai ketika aktor memilih fungsi *logout*. Kemudian sistem akan melakukan penghapusan *session* lalu menampilkan halaman *login*.

#### 4.6.3 Activity Diagram Melihat Jadwal Armada Bus

Aktivitas yang berjalan pada *use-case* melihat jadwal armada bus dapat dilihat pada Gambar 4.12. Aktivitas tersebut menggambarkan bagaimana proses melihat jadwal armada bus yang mengikuti proses pada *use-case scenario*

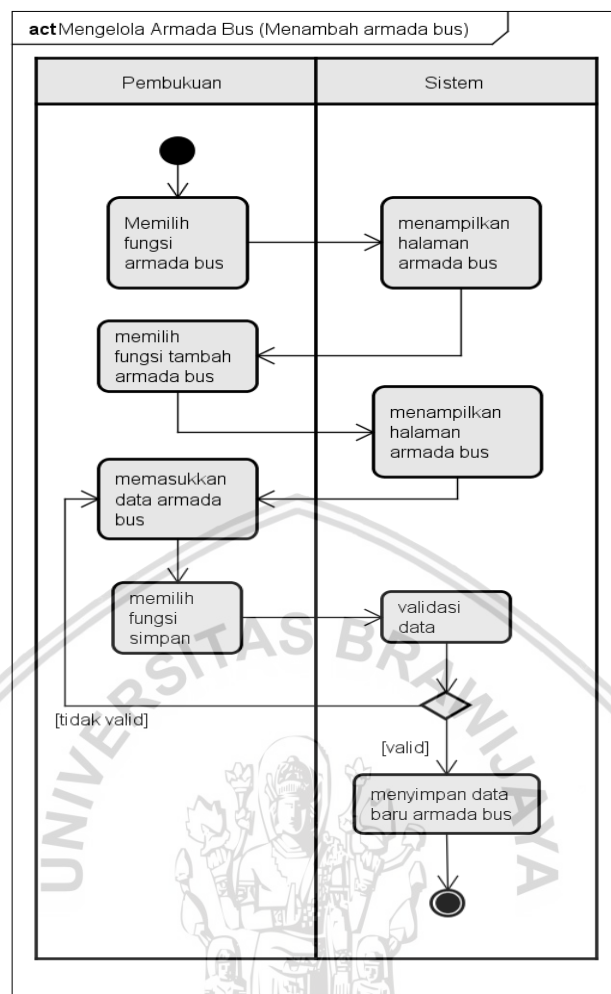


**Gambar 4.12 Activity Diagram Melihat Jadwal Armada**

Gambar 4.12 merupakan gambar *activity diagram* melihat jadwal armada bus. Aktivitas dimulai ketika aktor memilih fungsi jadwal bus lalu sistem akan menampilkan halaman jadwal bus. Kemudian aktor memasukkan rentang tanggal yang akan dicari dan memilih fungsi cari. Sistem akan menampilkan data jadwal armada bus.

#### 4.6.4 Activity Diagram Mengelola Armada Bus (Menambah Armada Bus)

Aktivitas yang berjalan pada usecase mengelola armada bus dapat dilihat pada Gambar 4.13. Aktivitas tersebut menggambarkan bagaimana proses menambah armada yang mengikuti proses pada *use-case scenario* mengelola armada bus

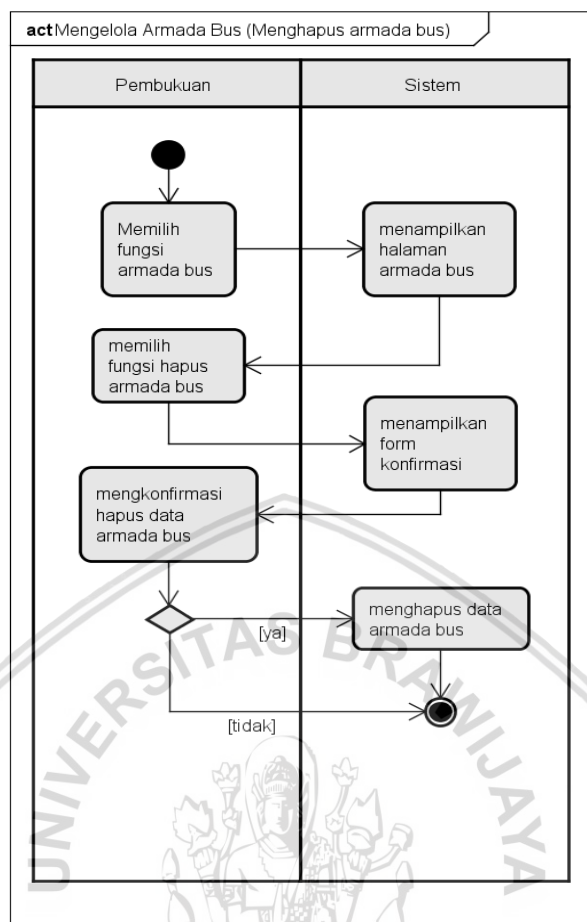


**Gambar 4.13 Activity Diagram Menambah Armada Bus**

Gambar 4.13 merupakan gambar *activity diagram* menambah armada bus. Aktivitas dimulai ketika aktor memilih fungsi armada bus dan sistem akan menampilkan halaman armada bus. Kemudian aktor memilih fungsi tambah armada dan sistem akan menampilkan halaman tambah armada bus. Aktor memasukkan data armada bus baru ke *form* yang telah disediakan dan memilih fungsi simpan. Sistem akan melakukan validasi terhadap data yang ingin disimpan, jadi ketika ada kesalahan aktor harus memperbaiki data yang ingin disimpan. Kemudian sistem akan menyimpan data armada bus baru.

#### 4.6.5 Activity Diagram Mengelola Armada Bus (Menghapus Armada Bus)

Aktivitas yang berjalan pada *use-case* mengelola armada bus dapat dilihat pada Gambar 4.14. Aktivitas tersebut menggambarkan bagaimana proses menghapus armada yang mengikuti proses pada *use-case scenario* mengelola armada bus



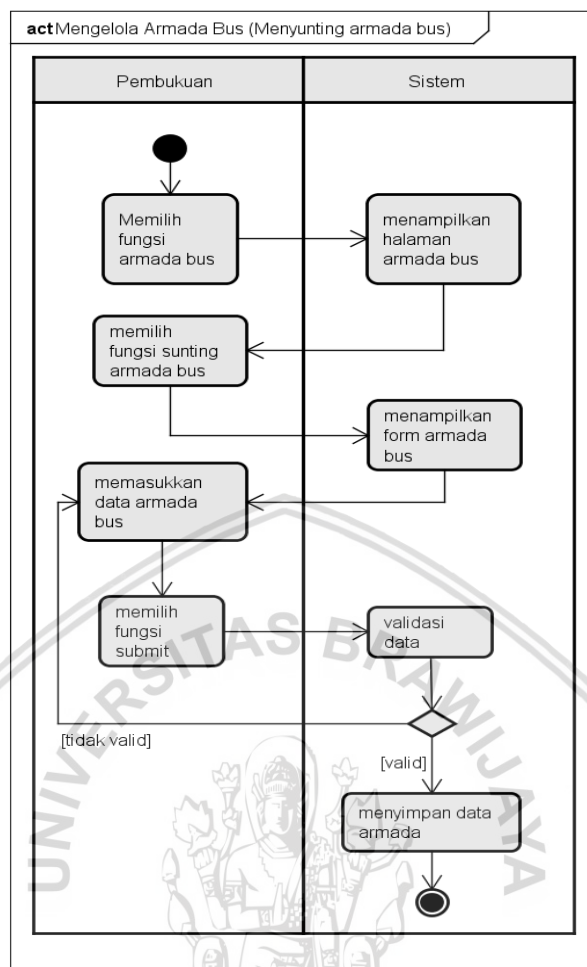
**Gambar 4.14 Activity Diagram Menghapus Armada Bus**

Gambar 4.14 merupakan gambar *activity diagram* menghapus armada bus. Aktivitas dimulai ketika aktor memilih fungsi armada bus dan sistem akan menampilkan halaman armada bus. Kemudian aktor memilih fungsi hapus armada bus dan sistem akan menampilkan *form* konfirmasi. Aktor mengkonfirmasi penghapusan data. Sistem akan menghapus data yang dipilih oleh aktor. Ketika aktor tidak mengkonfirmasi penghapusan, maka sistem tidak akan menghapus data yang dipilih aktor.

#### **4.6.6 Activity Diagram Mengelola Armada Bus (Menyunting Armada Bus)**

Aktivitas yang berjalan pada *use-case* mengelola armada bus dapat dilihat pada Gambar 4.15. Aktivitas tersebut menggambarkan bagaimana proses menyunting armada yang mengikuti proses pada *use-case scenario* mengelola armada bus



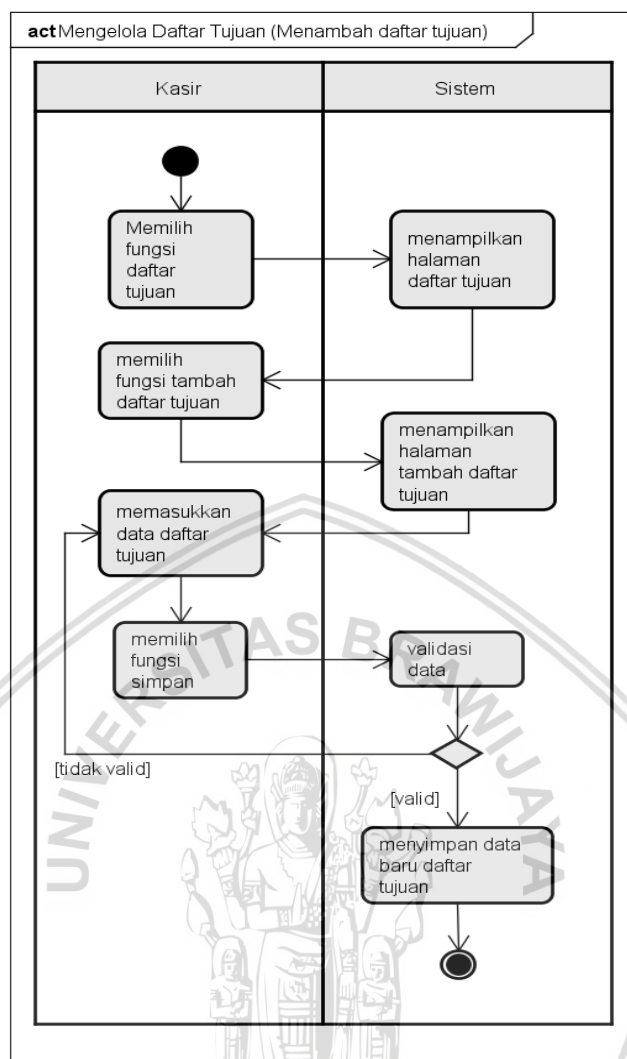


**Gambar 4.15 Activity Diagram Menyunting Armada Bus**

Gambar 4.15 merupakan gambar *activity diagram* menyunting armada bus. Aktivitas dimulai ketika aktor memilih fungsi armada bus dan sistem akan menampilkan halaman armada bus. Kemudian aktor memilih fungsi sunting armada dan sistem akan menampilkan halaman sunting armada bus. Aktor memasukkan data armada bus ke *form* yang telah disediakan dan memilih fungsi simpan. Sistem akan melakukan validasi terhadap data yang ingin disimpan, jadi ketika ada kesalahan aktor harus memperbaiki data yang ingin disimpan. Kemudian sistem akan menyimpan data armada bus yang disunting.

#### 4.6.7 Activity Diagram Mengelola Daftar Tujuan (Menambah Daftar Tujuan)

Aktivita yang berjalan pada *use-case* mengelola daftar tujuan dapat dilihat pada Gambar 4.16. Aktivitas tersebut menggambarkan bagaimana proses menambah daftar tujuan yang mengikuti proses pada *use-case scenario* mengelola daftar tujuan.

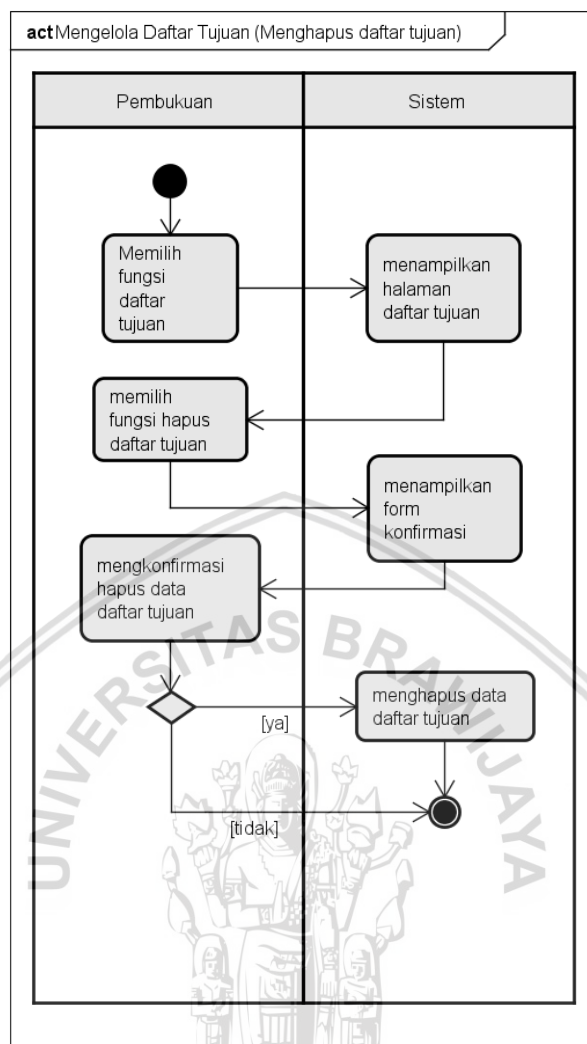


**Gambar 4.16 Activity Diagram Menambah Daftar Tujuan**

Gambar 4.16 merupakan gambar *activity diagram* menambah daftar tujuan. Aktivitas dimulai ketika aktor memilih fungsi daftar tujuan dan sistem akan menampilkan halaman daftar tujuan. Kemudian aktor memilih fungsi tambah daftar tujuan dan sistem akan menampilkan halaman tambah daftar tujuan. Aktor memasukkan data daftar tujuan baru ke *form* yang telah disediakan dan memilih fungsi simpan. Sistem akan melakukan validasi terhadap data yang ingin disimpan, jadi ketika ada kesalahan aktor harus memperbaiki data yang ingin disimpan. Kemudian sistem akan menyimpan data daftar tujuan baru.

#### 4.6.8 Activity Diagram Mengelola Daftar Tujuan (Menghapus Daftar Tujuan)

Aktivitas yang berjalan pada *use-case* mengelola daftar tujuan dapat dilihat pada Gambar 4.17. Aktivitas tersebut menggambarkan bagaimana proses menambah daftar tujuan yang mengikuti proses pada *use-case scenario* mengelola daftar tujuan.

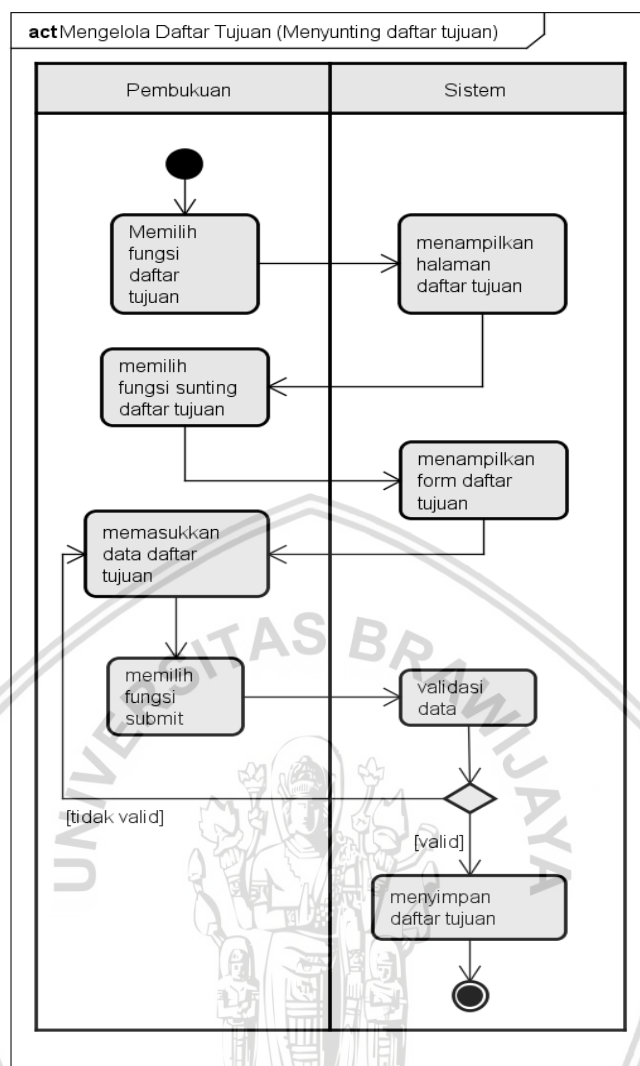


**Gambar 4.17 Activity Diagram Menghapus Daftar Tujuan**

Gambar 4.17 merupakan gambar *activity diagram* menghapus daftar tujuan. Aktivitas dimulai ketika aktor memilih fungsi daftar tujuan dan sistem akan menampilkan halaman daftar tujuan. Kemudian aktor memilih fungsi hapus daftar tujuan dan sistem akan menampilkan *form* konfirmasi. Aktor mengkonfirmasi penghapusan data. Sistem akan menghapus data yang dipilih oleh aktor. Ketika aktor tidak mengkonfirmasi penghapusan, maka sistem tidak akan menghapus data yang dipilih aktor.

#### **4.6.9 Activity Diagram Mengelola Daftar Tujuan (Menyunting Daftar Tujuan)**

Aktivitas yang berjalan pada *use-case* mengelola daftar tujuan dapat dilihat pada Gambar 4.18. Aktivitas tersebut menggambarkan bagaimana proses menambah daftar tujuan yang mengikuti proses pada *use-case scenario* mengelola daftar tujuan

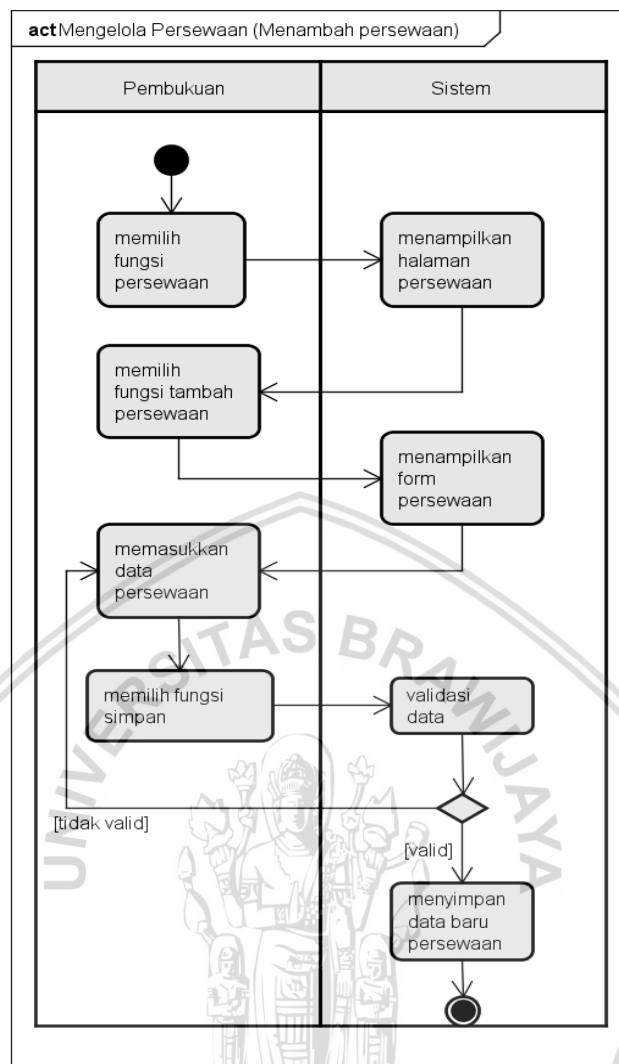


**Gambar 4.18 Activity Diagram Menyunting Daftar Tujuan**

Gambar 4.18 merupakan gambar *activity diagram* menyunting daftar tujuan. Aktivitas dimulai ketika aktor memilih fungsi daftar tujuan dan sistem akan menampilkan halaman daftar tujuan. Kemudian aktor memilih fungsi sunting daftar tujuan dan sistem akan menampilkan halaman sunting daftar tujuan. Aktor memasukkan data daftar tujuan ke *form* yang telah disediakan dan memilih fungsi simpan. Sistem akan melakukan validasi terhadap data yang ingin disimpan, jadi ketika ada kesalahan aktor harus memperbaiki data yang ingin disimpan. Kemudian sistem akan menyimpan data daftar tujuan yang disunting.

#### 4.6.10 Activity Diagram Mengelola Persewaan (Menambah Persewaan)

Aktivitas yang berjalan pada *use-case* mengelola persewaan dapat dilihat pada Gambar 4.19. Aktivitas tersebut menggambarkan bagaimana proses menambah persewaan yang mengikuti proses pada *use-case scenario* mengelola persewaan

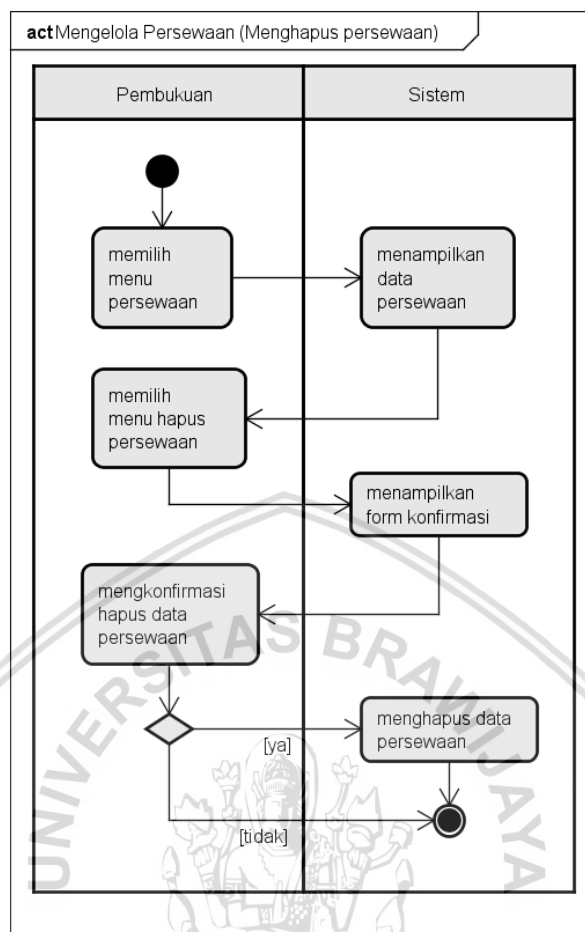


**Gambar 4.19 Activity Diagram Menambah Persewaan**

Gambar 4.19 merupakan gambar *activity diagram* menambah persewaan. Aktivitas dimulai ketika aktor memilih fungsi persewaan dan sistem akan menampilkan halaman persewaan. Kemudian aktor memilih fungsi tambah persewaan dan sistem akan menampilkan halaman tambah persewaan. Aktor memasukkan data persewaan baru ke *form* yang telah disediakan dan memilih fungsi simpan. Sistem akan melakukan validasi terhadap data yang ingin disimpan, jadi ketika ada kesalahan aktor harus memperbaiki data yang ingin disimpan. Kemudian sistem akan menyimpan data persewaan baru.

#### 4.6.11 Activity Diagram Mengelola Persewaan (Menghapus Persewaan)

Aktivitas yang berjalan pada *use-case* mengelola persewaan dapat dilihat pada Gambar 4.17. Aktivitas tersebut menggambarkan bagaimana proses menghapus persewaan yang mengikuti proses pada *use-case scenario* mengelola persewaan



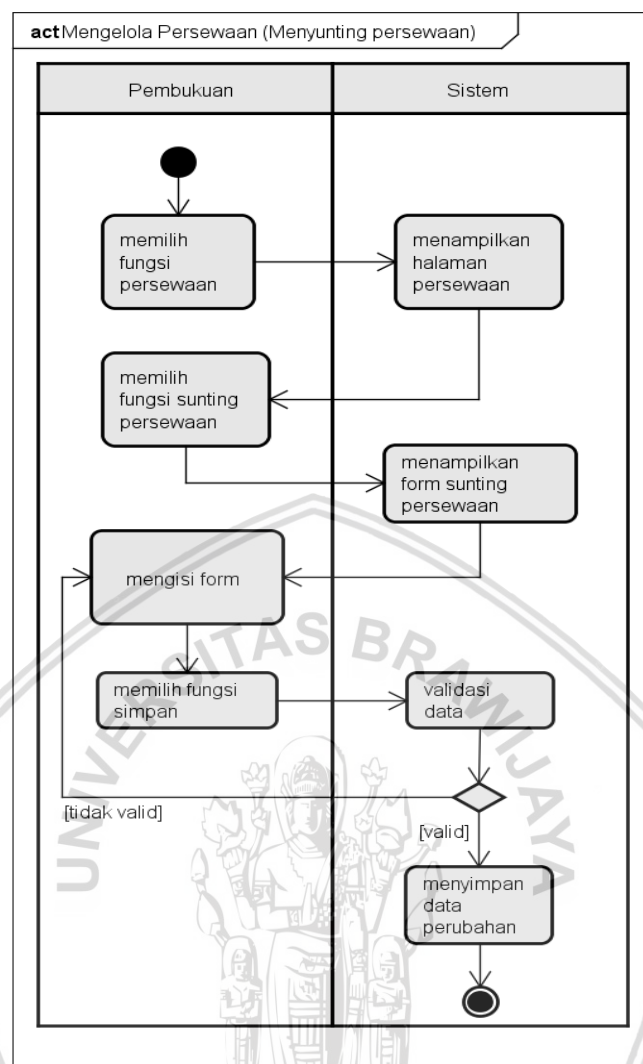
**Gambar 4.20 Activity Diagram Menghapus Persewaan**

Gambar 4.20 merupakan gambar *activity diagram* menghapus persewaan. Aktivitas dimulai ketika aktor memilih fungsi persewaan dan sistem akan menampilkan halaman persewaan. Kemudian aktor memilih fungsi hapus persewaan dan sistem akan menampilkan *form* konfirmasi. Aktor mengkonfirmasi penghapusan data. Sistem akan menghapus data yang dipilih oleh aktor. Ketika aktor tidak mengkonfirmasi penghapusan, maka sistem tidak akan menghapus data yang dipilih aktor.

#### 4.6.12 Activity Diagram Mengelola Persewaan (Menyunting Persewaan)

Aktivitas yang berjalan pada *use-case* mengelola persewaan dapat dilihat pada Gambar 4.21. Aktivitas tersebut menggambarkan bagaimana proses menyunting persewaan yang mengikuti proses pada *use-case scenario* mengelola persewaan



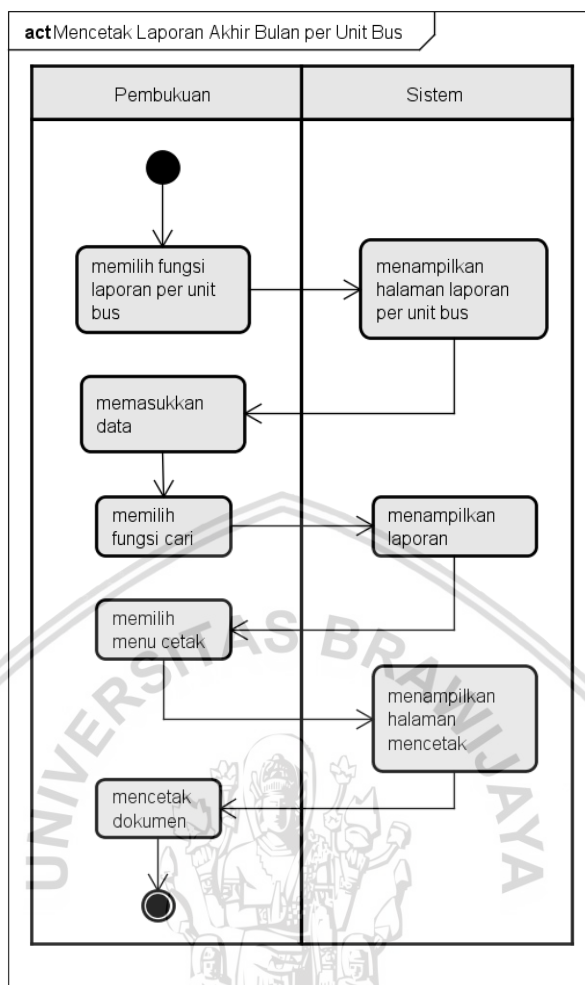


**Gambar 4.21 Activity Diagram Menyunting Persewaan**

Gambar 4.21 merupakan gambar *activity diagram* menyunting persewaan. Aktivitas dimulai ketika aktor memilih fungsi persewaan dan sistem akan menampilkan halaman persewaan. Kemudian aktor memilih fungsi sunting persewaan dan sistem akan menampilkan halaman sunting persewaan. Aktor memasukkan data persewaan ke *form* yang telah disediakan dan memilih fungsi simpan. Sistem akan melakukan validasi terhadap data yang ingin disimpan, jadi ketika ada kesalahan aktor harus memperbaiki data yang ingin disimpan. Kemudian sistem akan menyimpan data persewaan yang disunting.

#### 4.6.13 Activity Diagram Mencetak Laporan Akhir Bulan per Unit Bus

Aktivitas yang berjalan pada *use-case* mencetak laporan akhir bulan per unit bus dapat dilihat pada Gambar 4.19. Aktivitas tersebut menggambarkan bagaimana proses *login* yang mengikuti proses pada *use-case scenario* mencetak laporan akhir bulan per unit bus.

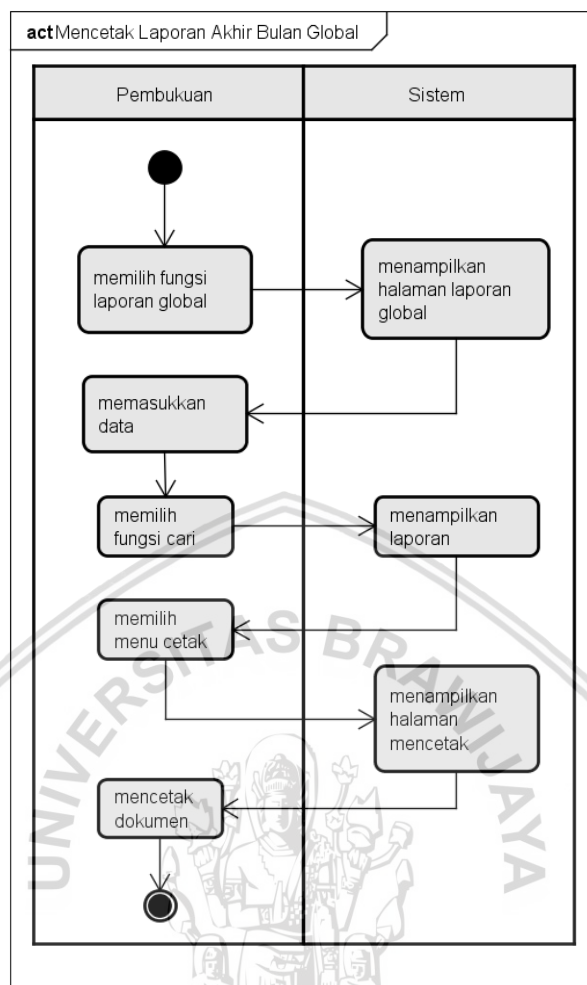


**Gambar 4.22 Activity Diagram Mencetak Laporan Akhir Bulan per Unit**

Gambar 4.21 merupakan gambar *activity diagram* mencetak laporan akhir bulan per unit. Aktivitas dimulai ketika aktor memilih fungsi laporan per unit bus dan sistem akan menampilkan halaman laporan unit. Kemudian aktor memasukkan periode dan nomor polisi dan memilih fungsi cari. Lalu sistem akan menampilkan laporan pada periode untuk unit sesuai dengan masukkan aktor. Selanjutnya aktor memilih fungsi cetak dan sistem akan menampilkan halaman cetak dokumen laporan tersebut.

#### **4.6.14 Activity Diagram Mencetak Laporan Akhir Bulan Global**

Aktivitas yang berjalan pada *use-case* mencetak laporan akhir bulan global dapat dilihat pada Gambar 4.23. Aktivitas tersebut menggambarkan bagaimana proses *login* yang mengikuti proses pada *use-case scenario* mencetak laporan akhir bulan global.

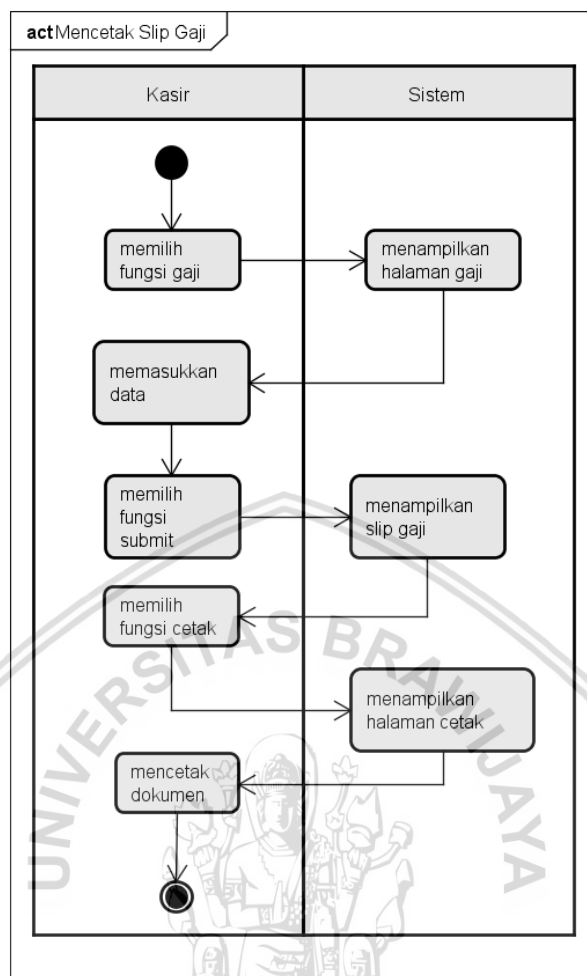


**Gambar 4.23 Activity Diagram Laporan Akhir Bulan Global**

Gambar 4.23 merupakan gambar *activity diagram* mencetak laporan akhir bulan global. Aktivitas dimulai ketika aktor memilih fungsi laporan global dan sistem akan menampilkan halaman laporan global. Kemudian aktor memasukkan periode dan memilih fungsi cari. Lalu sistem akan menampilkan laporan pada periode sesuai dengan masukkan aktor. Selanjutnya aktor memilih fungsi cetak dan sistem akan menampilkan halaman cetak dokumen laporan tersebut.

#### 4.6.15 Activity Diagram Mencetak Slip Gaji

Aktivitas yang berjalan pada *use-case* mencetak slip gaji dapat dilihat pada Gambar 4.24. Aktivitas tersebut menggambarkan bagaimana proses *login* yang mengikuti proses pada *use-case scenario* mencetak slip gaji

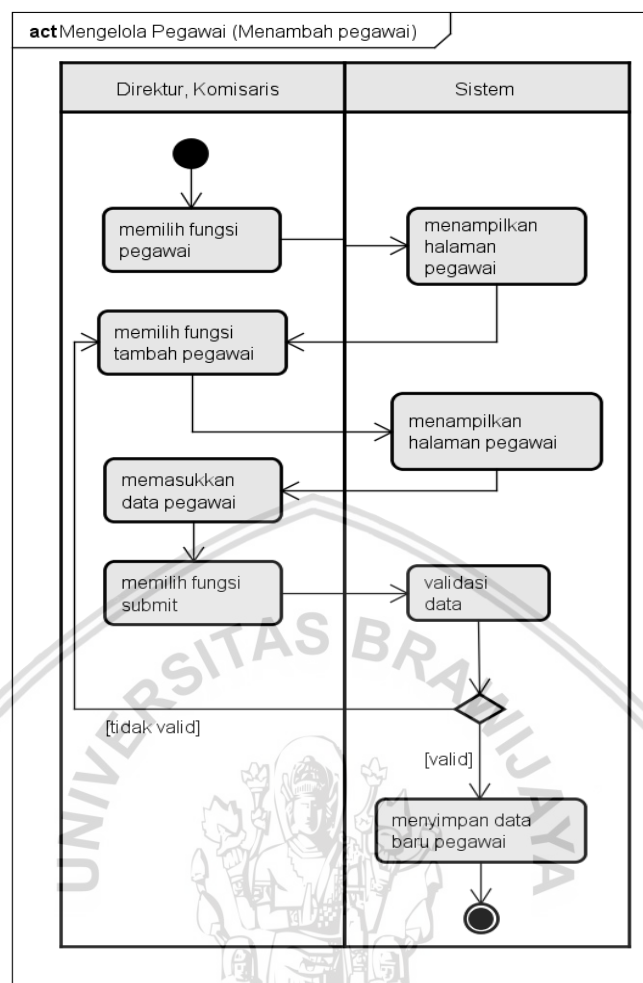


**Gambar 4.24 Activity Diagram Mencetak Slip Gaji**

Gambar 4.24 merupakan gambar *activity diagram* mencetak slip gaji. Aktivitas dimulai ketika aktor memilih fungsi gaji dan sistem akan menampilkan halaman gaji. Kemudian aktor memasukkan data dan memilih fungsi cari. Lalu sistem akan menampilkan halaman slip gaji sesuai dengan masukkan aktor. Selanjutnya aktor memilih fungsi cetak dan sistem akan menampilkan halaman cetak dokumen slip gaji tersebut.

#### **4.6.16 Activity Diagram Mengelola Pegawai (Menambah Pegawai)**

Aktivitas yang berjalan pada *use-case* mengelola pegawai dapat dilihat pada Gambar 4.25. Aktivitas tersebut menggambarkan bagaimana proses menambah pegawai yang mengikuti proses pada *use-case scenario* mengelola pegawai

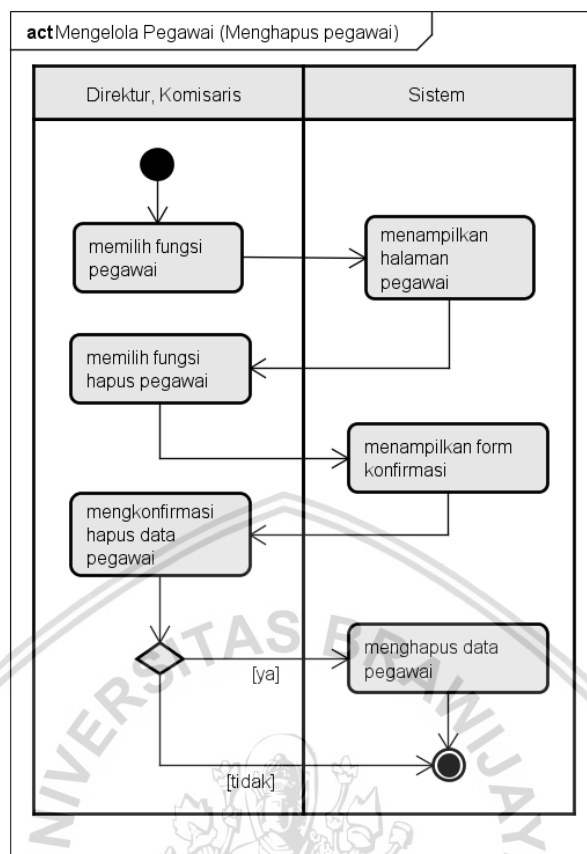


**Gambar 4.25 Activity Diagram Menambah Pegawai**

Gambar 4.25 merupakan gambar *activity diagram* menambah pegawai. Aktivitas dimulai ketika aktor memilih fungsi pegawai dan sistem akan menampilkan halaman pegawai. Kemudian aktor memilih fungsi tambah pegawai dan sistem akan menampilkan halaman tambah pegawai. Aktor memasukkan data pegawai baru ke *form* yang telah disediakan dan memilih fungsi simpan. Sistem akan melakukan validasi terhadap data yang ingin disimpan, jadi ketika ada kesalahan aktor harus memperbaiki data yang ingin disimpan. Kemudian sistem akan menyimpan data pegawai baru.

#### 4.6.17 Activity Diagram Mengelola Pegawai (Menghapus Pegawai)

Aktivitas yang berjalan pada *use-case* mengelola pegawai dapat dilihat pada Gambar 4.26. Aktivitas tersebut menggambarkan bagaimana proses menghapus pegawai yang mengikuti proses pada *use-case scenario* mengelola pegawai.



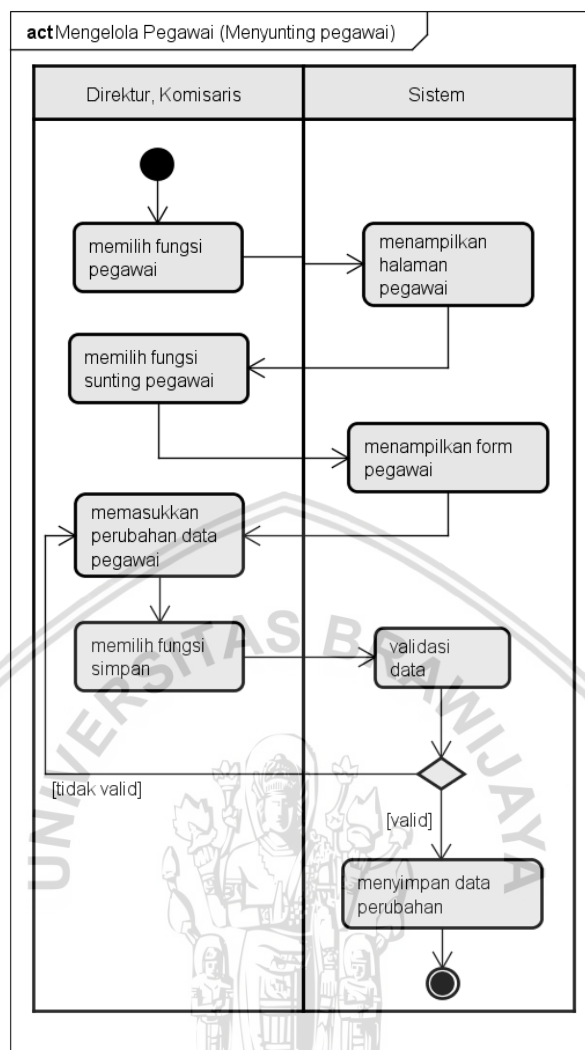
**Gambar 4.26 Activity Diagram Menghapus Pegawai**

Gambar 4.26 merupakan gambar *activity diagram* menghapus pegawai. Aktivitas dimulai ketika aktor memilih fungsi pegawai dan sistem akan menampilkan halaman pegawai. Kemudian aktor memilih fungsi hapus pegawai dan sistem akan menampilkan *form* konfirmasi. Aktor mengkonfirmasi penghapusan data. Sistem akan menghapus data yang dipilih oleh aktor. Ketika aktor tidak mengkonfirmasi penghapusan, maka sistem tidak akan menghapus data yang dipilih aktor.

#### **4.6.18 Activity Diagram Mengelola Pegawai (Menyunting Pegawai)**

Aktivitas yang berjalan pada *use-case* mengelola pegawai dapat dilihat pada Gambar 4.27. Aktivitas tersebut menggambarkan bagaimana proses menyunting pegawai yang mengikuti proses pada *use-case scenario* mengelola pegawai.



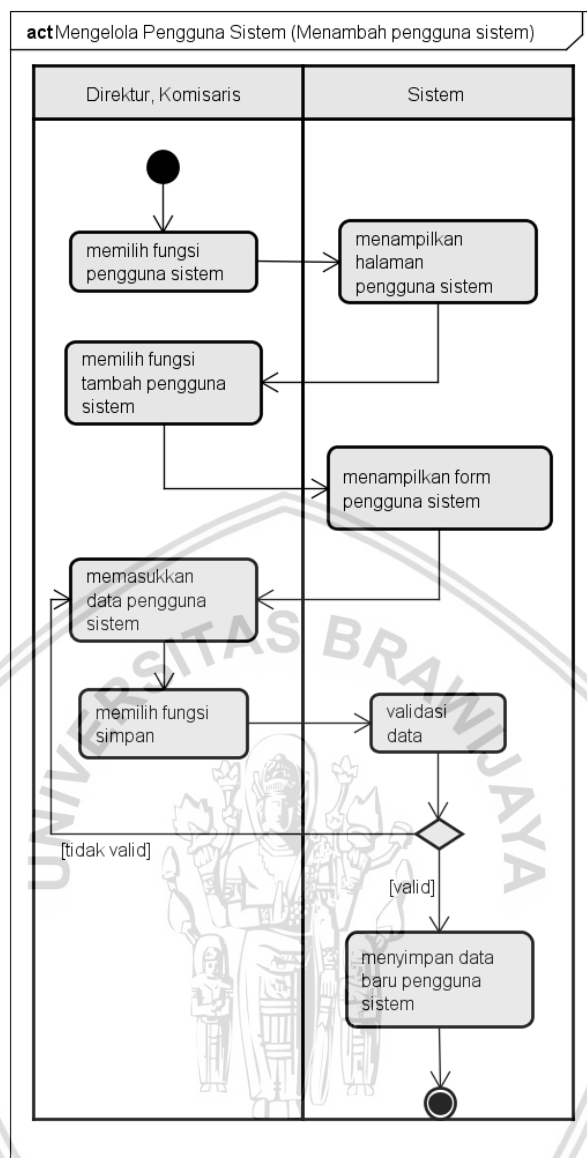


**Gambar 4.27 Activity Diagram Menyunting Pegawai**

Gambar 4.27 merupakan gambar *activity diagram* menyunting pegawai. Aktivitas dimulai ketika aktor memilih fungsi pegawai dan sistem akan menampilkan halaman pegawai. Kemudian aktor memilih fungsi sunting pegawai dan sistem akan menampilkan halaman sunting pegawai. Aktor memasukkan data perubahan ke *form* yang telah disediakan dan memilih fungsi simpan. Sistem akan melakukan validasi terhadap data yang ingin disimpan, jadi ketika ada kesalahan aktor harus memperbaiki data yang ingin disimpan. Kemudian sistem akan menyimpan data pegawai yang disunting.

#### 4.6.19 Activity Diagram Mengelola Pengguna Sistem (Menambah Pengguna Sistem)

Aktivitas yang berjalan pada *use-case* mengelola pengguna sistem dapat dilihat pada Gambar 4.28. Aktivitas tersebut menggambarkan bagaimana proses menambah pengguna sistem yang mengikuti proses pada *use-case scenario* mengelola pengguna sistem.

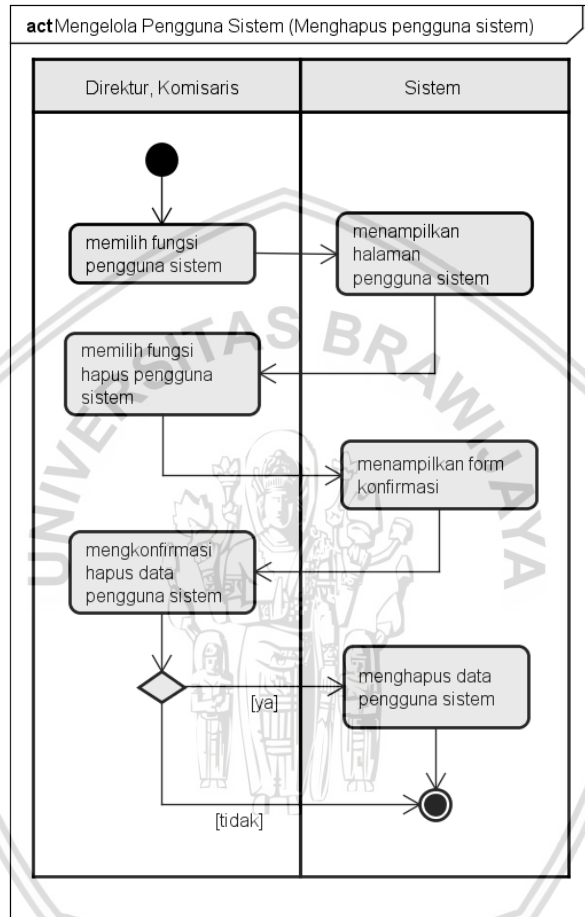


**Gambar 4.28 Activity Diagram Menambah Pengguna Sistem**

Gambar 4.28 merupakan gambar *activity diagram* menambah pengguna sistem. Aktivitas dimulai ketika aktor memilih fungsi pengguna sistem dan sistem akan menampilkan halaman pengguna sistem. Kemudian aktor memilih fungsi tambah pengguna sistem dan sistem akan menampilkan halaman tambah pengguna sistem. Aktor memasukkan data pengguna sistem baru ke *form* yang telah disediakan dan memilih fungsi simpan. Sistem akan melakukan validasi terhadap data yang ingin disimpan, jadi ketika ada kesalahan aktor harus memperbaiki data yang ingin disimpan. Kemudian sistem akan menyimpan data pengguna sistem baru.

#### 4.6.20 Activity Diagram Mengelola Pengguna Sistem (Menghapus Pengguna Sistem)

Aktivitas yang berjalan pada *use-case* mengelola pengguna sistem dapat dilihat pada Gambar 4.29. Aktivitas tersebut menggambarkan bagaimana proses menghapus pengguna sistem yang mengikuti proses pada *use-case scenario* mengelola pengguna sistem

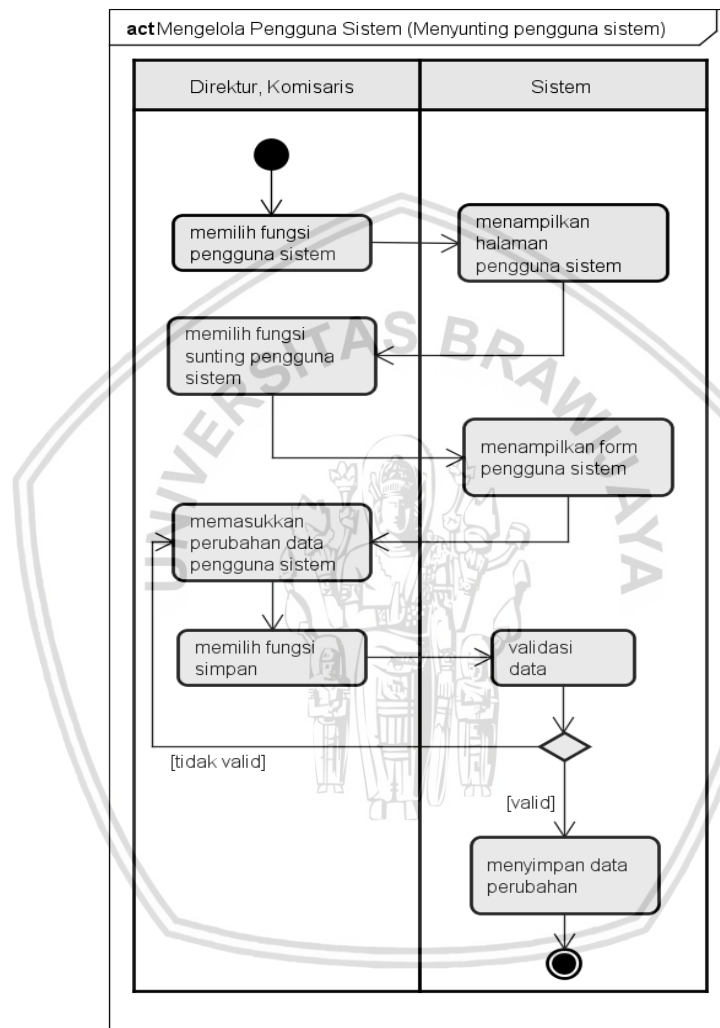


**Gambar 4.29 Activity Diagram Menghapus Pengguna Sistem**

Gambar 4.29 merupakan gambar *activity diagram* menghapus pengguna sistem. Aktivitas dimulai ketika aktor memilih fungsi pengguna sistem dan sistem akan menampilkan halaman pengguna sistem. Kemudian aktor memilih fungsi hapus pengguna sistem dan sistem akan menampilkan *form* konfirmasi. Aktor mengkonfirmasi penghapusan data. Sistem akan menghapus data yang dipilih oleh aktor. Ketika aktor tidak mengkonfirmasi penghapusan, maka sistem tidak akan menghapus data yang dipilih aktor.

#### 4.6.21 Activity Diagram Mengelola Pengguna Sistem (Menyunting Pengguna Sistem)

Aktivitas yang berjalan pada *use-case* mengelola pengguna sistem dapat dilihat pada Gambar 4.30. Aktivitas tersebut menggambarkan bagaimana proses menyunting pengguna sistem yang mengikuti proses pada *use-case scenario* mengelola pengguna sistem

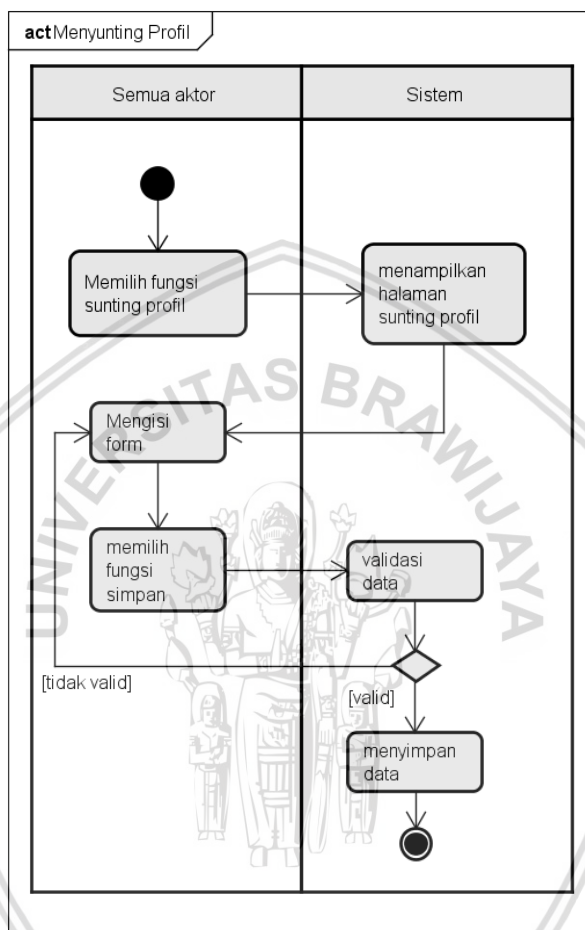


**Gambar 4.30 Activity Diagram Menyunting Pengguna Sistem**

Gambar 4.30 merupakan gambar *activity diagram* menyunting pengguna sistem. Aktivitas dimulai ketika aktor memilih fungsi pengguna sistem dan sistem akan menampilkan halaman pengguna sistem. Kemudian aktor memilih fungsi sunting pengguna sistem dan sistem akan menampilkan halaman sunting pengguna sistem. Aktor memasukkan data persewaan ke *form* yang telah disediakan dan memilih fungsi simpan. Sistem akan melakukan validasi terhadap data yang ingin disimpan, jadi ketika ada kesalahan aktor harus memperbaiki data yang ingin disimpan. Kemudian sistem akan menyimpan data pengguna sistem yang disunting.

#### 4.6.22 Activity Diagram Menyunting Profil

Aktivitas yang berjalan pada *use-case* menyunting profil dapat dilihat pada Gambar 4.31. Aktivitas tersebut menggambarkan bagaimana proses menyunting profil yang mengikuti proses pada *use-case scenario* menyunting profil.



**Gambar 4.31 Activity Diagram Menyunting Profil**

Gambar 4.31 merupakan gambar *activity diagram* menyunting profil. Aktivitas dimulai ketika aktor memilih fungsi *my profil* dan sistem akan menampilkan halaman sunting profil. Sistem akan menampilkan halaman sunting pengguna sistem. Aktor memasukkan perubahan data ke *form* yang telah disediakan dan memilih fungsi simpan. Sistem akan melakukan validasi terhadap data yang ingin disimpan, jadi ketika ada kesalahan aktor harus memperbaiki data yang ingin disimpan. Kemudian sistem akan menyimpan data profil yang disunting.

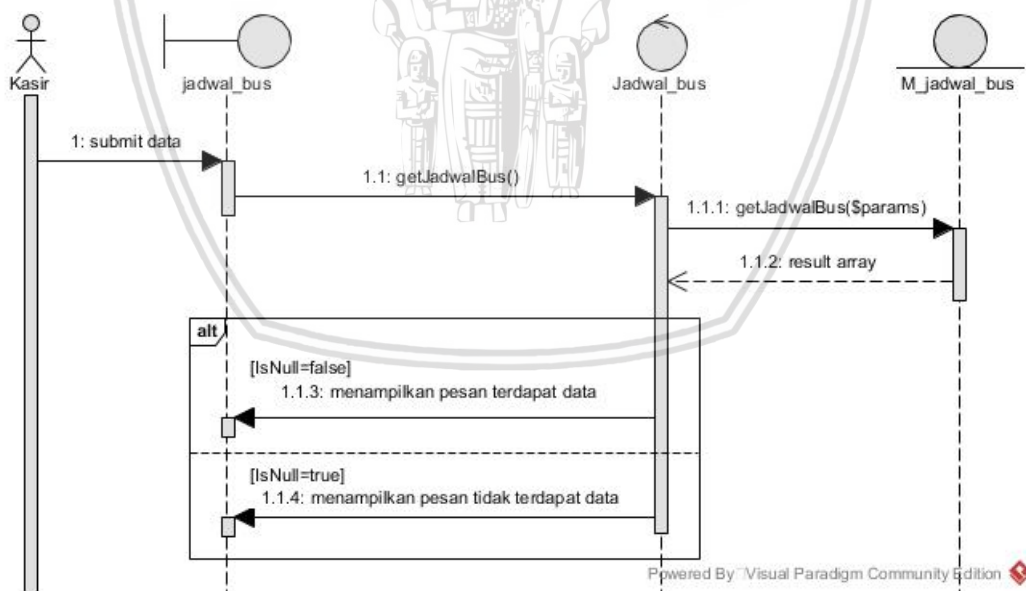
## BAB 5 PERANCANGAN

### 5.1 Sequence diagram

Pada bagian ini digambarkan mengenai alur proses yang terjadi di dalam sistem berdasarkan urutan waktu (sekuensial). Perancangan *Sequence diagram* berdasarkan pada masing-masing *scenario use-case* yang telah didefinisikan pada bab sebelumnya. Berikut beberapa *Sequence diagram* yang dirancang pada penelitian ini.

#### 5.1.1 Melihat Jadwal Bus

*Sequence diagram* yang terdapat dalam Gambar 5.1 merupakan visualisasi interaksi antar objek pada proses melihat jadwal bus. Proses ini dimulai ketika aktor kasir melakukan *submit* data pada *object boundary* jadwal\_bus. Kemudian *boundary* jadwal\_bus memanggil *function* *getJadwalBus()* pada *object control* Jadwal\_bus. Data yang telah dikirim oleh aktor kasir dikirimkan ke *object control* tersebut. Lalu *object control* Jadwal\_bus memanggil *function* *getJadwalBus(\$params)* pada *object entity* M\_jadwal\_bus. Pemanggilan *function* pada *object entity* menggunakan *argument* yang datanya berasal dari data yang telah dikirim oleh aktor kasir. Selanjutnya, setelah *object entity* M\_jadwal\_bus dipanggil akan memberikan balasan kepada jadwal bus kepada *object control* Jadwal\_bus berupa data yang telah didapatkan pada *database*. Lalu *object control* tersebut mengirimkan data pada *object boundary* jadwal\_bus.



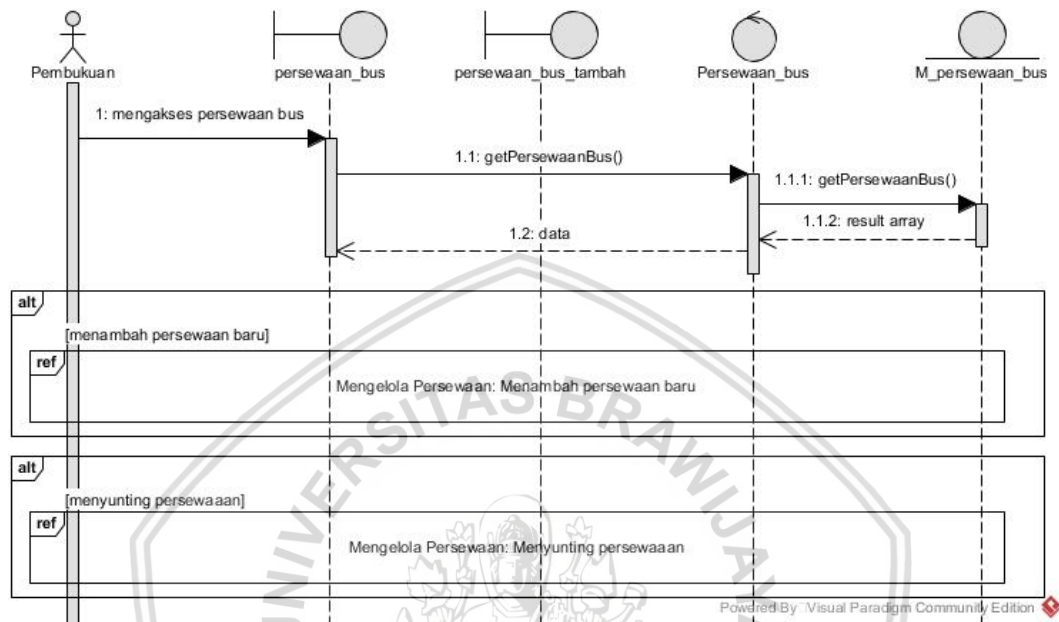
Gambar 5.1 *Sequence diagram* Melihat Jadwal Bus

#### 5.1.2 Mengelola Persewaan

*Sequence diagram* yang terdapat dalam Gambar 5.2 merupakan visualisasi interaksi antar objek pada proses mengelola persewaan. Proses ini dimulai ketika aktor pembukuan mengakses *boundary* persewaan\_bus. Pada saat *boundary*



tersebut diakses akan melakukan pemanggilan *function* *getPersewaanBus()* pada *object control* *persewaan\_bus*. Lalu *object control* tersebut akan memanggil *object entity* *M\_persewaan\_bus*. Setelah *object entity* tersebut dipanggil akan memberikan balasan pada *object control* berupa data yang telah diambil dari *database*. Kemudian *object control* mengirim data yang telah didapatkan dari *object entity*.

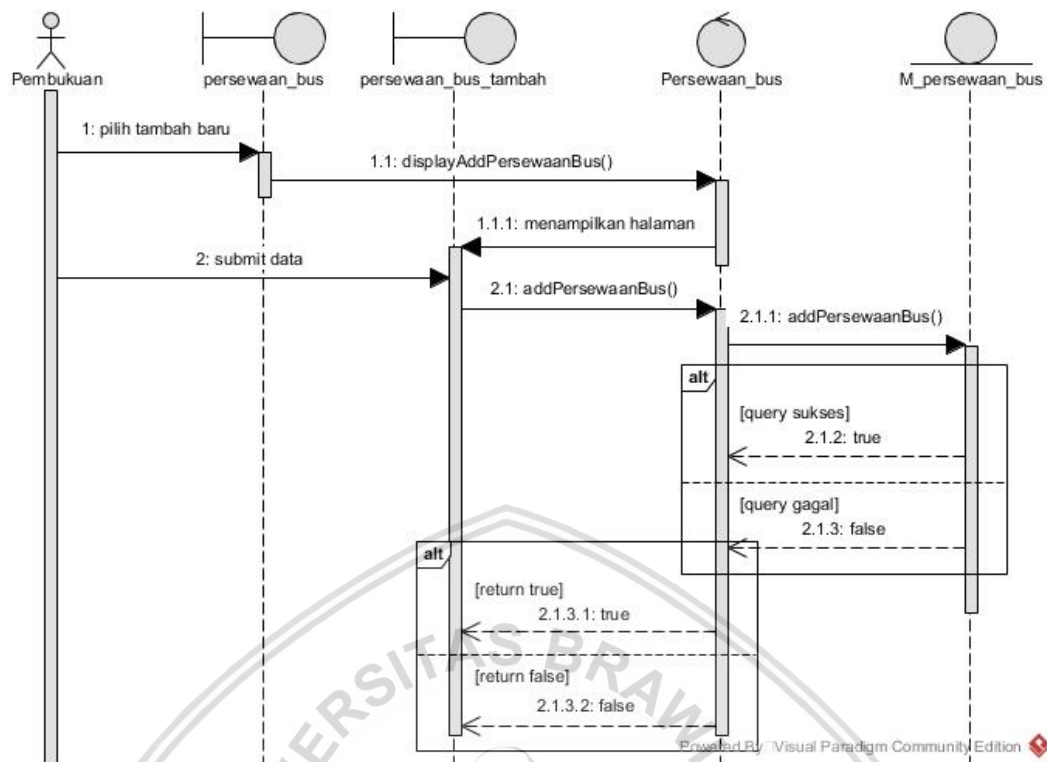


Gambar 5.2 Sequence Diagram Mengelola Persewaan

#### 4.1.1.1 Mengelola Persewaan: Menambah Persewaan

*Sequence diagram* yang terdapat dalam Gambar 5.3 merupakan visualisasi interaksi antar objek pada proses menambah persewaan dari alternatif proses mengelola persewaan. Proses ini dimulai ketika aktor pembukuan memilih tambah baru. Saat memilih tambah baru akan memanggil *function* *displayAddPersewaanBus()* yang terdapat pada *object control* *Persewaan\_bus*. Kemudian *function* tersebut akan menampilkan halaman *persewaan\_bus\_tambah*.

Setelah tampil halaman untuk menambah data persewaan, aktor pembukuan akan melakukan *submit* data. Kemudian akan memanggil *function* *addPersewaan* pada *object control* *Persewaan\_bus*. Lalu memanggil *function* *addPersewaanBus()* pada *object entity* *M\_persewaan\_bus*. Fungsi dari *function* *addPersewaanBus* pada *object entity* *M\_persewaan\_bus* yaitu untuk menyimpan data ke dalam *database* dari data yang telah dimasukkan oleh aktor pembukuan. Ketika *object entity* dipanggil akan memberikan balasan berupa *true* atau *false*. Balasan *true* ketika data berhasil masuk ke dalam *database*, sedangkan untuk balasan *false* ketika data gagal dimasukkan ke dalam *database*. Setelah *object entity* memberikan balasan, selanjutnya *object control* juga memberikan balasan kepada *object boundary* berupa *true* atau *false* sesuai dengan balasan dari *object entity*.

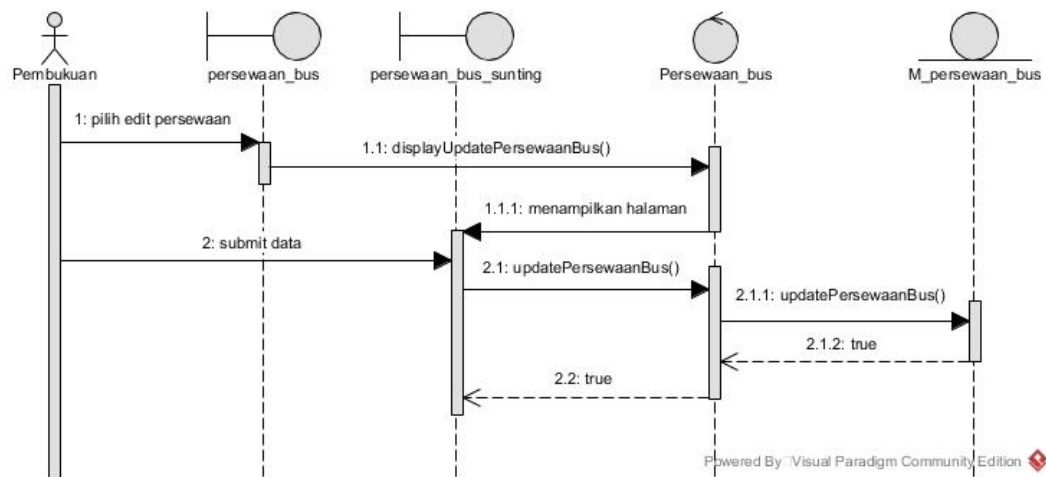


**Gambar 5.3 Sequence Diagram Mengelola Persewaan: Menambah Persewaan**

#### 4.1.1.2 Mengelola Persewaan: Menyunting Persewaan

*Sequence diagram* yang terdapat dalam Gambar 5.4 merupakan visualisasi interaksi antar objek pada proses menyunting persewaan dari alternatif proses mengelola persewaan. Proses ini dimulai ketika aktor pembukuan memilih edit persewaan pada *object boundary* persewaan\_bus. Saat memilih edit persewaan akan memanggil *function* `displayUpdatePersewaanBus()` yang terdapat pada *object control* Persewaan\_bus. Kemudian *function* tersebut akan menampilkan halaman persewaan\_bus\_sunting.

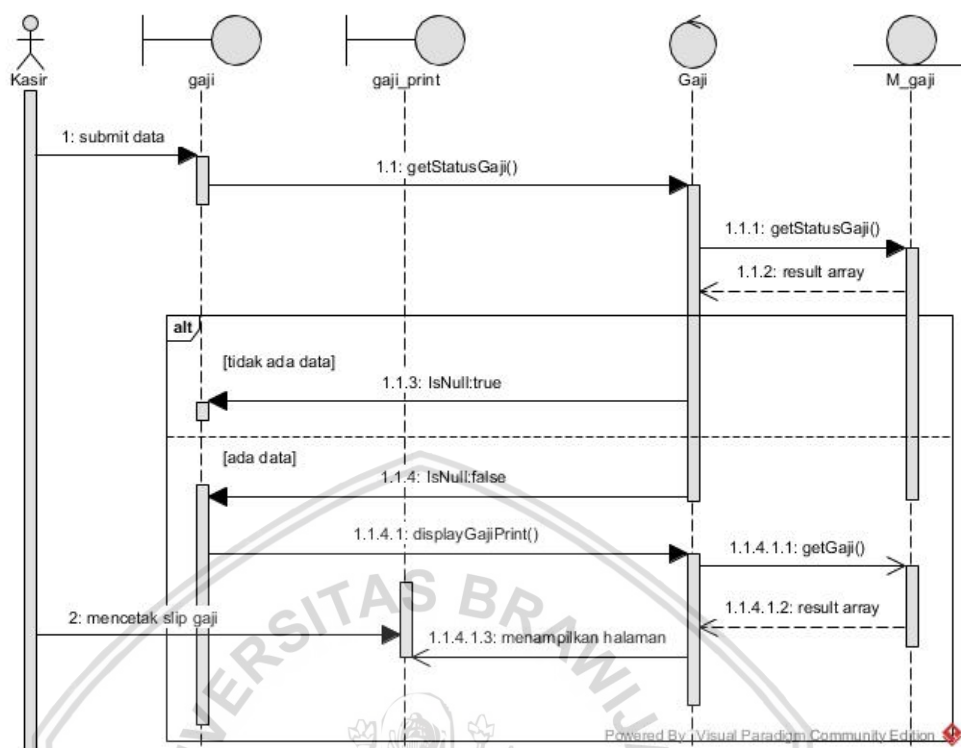
Setelah tampil halaman untuk sunting data persewaan, aktor pembukuan akan melakukan *submit* data. Ketika melakukan *submit* data akan memanggil *function* `updatePersewaanBus()` pada *object control* PersewaanBus. Kemudian akan memanggil *function* `updatePersewaanBus()` pada *object entity* M\_persewaan\_bus. Fungsi dari *function* `updatePersewaanBus` pada *object entity* M\_persewaan\_bus yaitu untuk menyimpan perubahan data ke dalam *database* dari data yang telah dimasukkan oleh aktor pembukuan. Ketika *object entity* dipanggil akan memberikan balasan berupa nilai *true*. Setelah *object entity* memberikan balasan, selanjutnya *object control* juga memberikan balasan kepada *object boundary* berupa nilai *true*.



**Gambar 5.4 Sequence Diagram Mengelola Persewaan: Menyunting Persewaan**

### 5.1.3 Cetak Slip Gaji

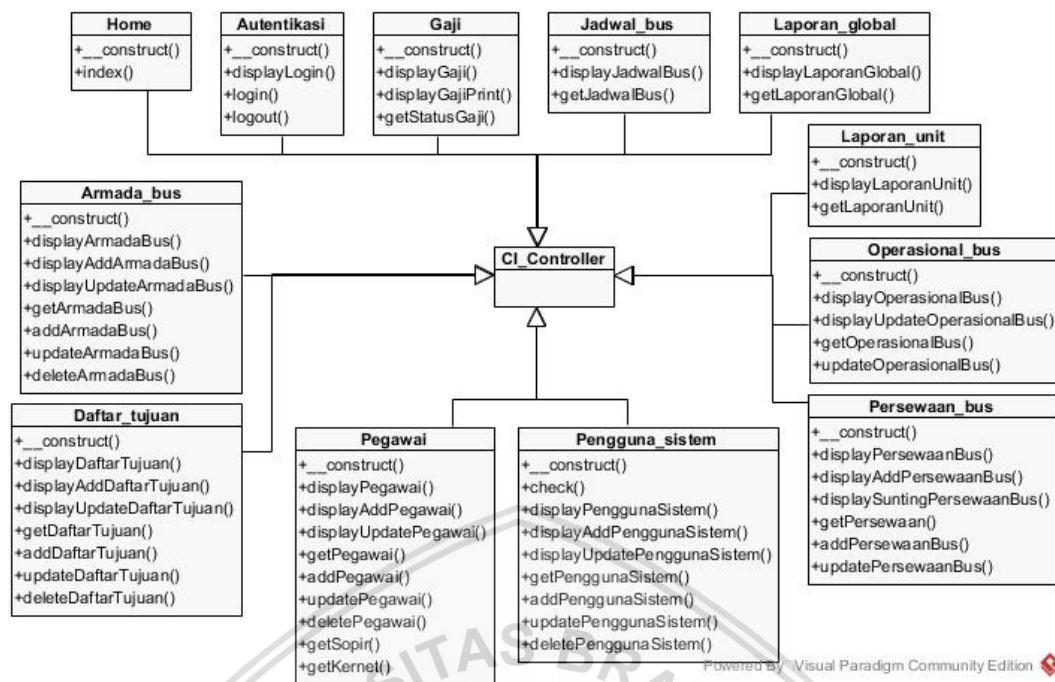
*Sequence diagram* yang terdapat dalam Gambar 5.5 merupakan visualisasi interaksi antar objek pada proses mencetak slip gaji. Proses ini dimulai ketika aktor kasir melakukan *submit* data pada *object boundary* gaji. Ketika melakukan *submit* akan memanggil *function* *getStatusGaji()* yang terdapat pada *object control* Gaji. Kemudian akan memanggil *function* *getStatusGaji()* pada *object entity* M\_gaji. Kemudian akan memberikan balasan berupa data yang didapat dari *database*. Pada *object control* setelah melakukan pemanggilan *function* pada *object entity* akan memberikan balasan kepada *object boundary* gaji. Balasan yang diberikan berupa nilai *true* atau *false*. Nilai *true* ketika data ada, sedangkan nilai *false* ketika data tidak ada. Ketika ada data akan memanggil *function* *displayGajiPrint()* pada *object control*. Lalu memanggil *function* *getGaji()* pada *object entity* dan *object entity* akan memberikan balasan berupa data yang diambil dari *database*. Setelah itu *object control* akan menampilkan halaman *gaji\_print* dan aktor melakukan cetak slip gaji.



Gambar 5.5 Sequence Diagram Cetak Slip Gaji

## 4.2 Class Diagram

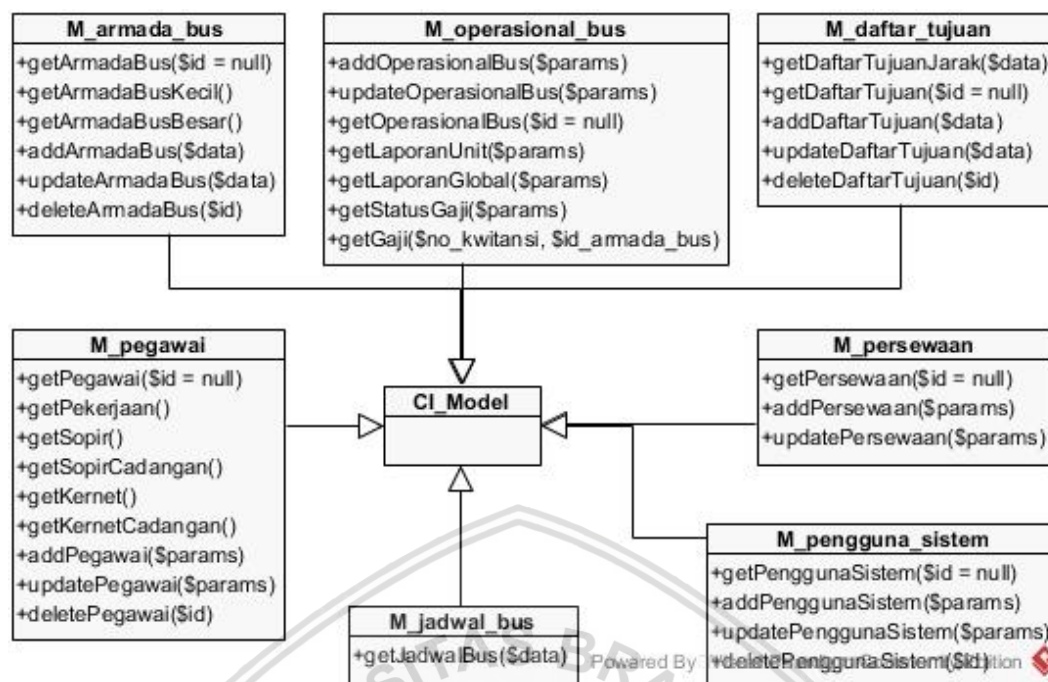
Setelah diketahui mengenai alur proses pada *use-case* yang telah dijelaskan pada sub bab 5.1 yaitu *sequence diagram*, langkah berikutnya adalah mengetahui hubungan antar *class* yang disertai dengan *atribut* dan *operan(method/function)*. Sehingga mempermudah dalam pembuatan perangkat lunak mengenai *class*, *attribute* dan *function* apa saja yang harus dibuat hingga hubungan antar *class* tersebut dengan *class* lainnya. Hal tersebut akan dijelaskan dengan membuat *class diagram* yang akan ditunjukkan pada gambar 5.6 dan gambar 5.7.



**Gambar 5.6 Class Diagram sebagai Logical Class**

Pada gambar 5.6 dijelaskan *class* yang akan digunakan untuk perancangan perangkat lunak yaitu terdapat 13 *class*. *Class* Home digunakan untuk menangani tampilan pada halaman utama. *Class* Autentikasi digunakan untuk menangani tampilan pada halaman *login* dan beberapa *function* untuk proses autentikasi. *Class* Gaji digunakan untuk menangani halaman gaji dan beberapa *function* pengelolaan data gaji. *Class* Jadwal\_bus digunakan untuk menangani halaman jadwal bus dan beberapa *function* pengelolaan data jadwal bus. *Class* Laporan\_global digunakan untuk menangani halaman laporan global dan beberapa *function* pengelolaan data laporan global. *Class* Laporan\_unit digunakan untuk menangani halaman laporan global dan beberapa *function* pengelolaan data laporan unit. *Class* Operasional\_bus digunakan untuk menangani halaman operasional bus dan beberapa *function* pengelolaan data operasional bus. *Class* Persewaan\_bus digunakan untuk menangani halaman persewaan bus dan beberapa *function* pengelolaan data persewaan bus. *Class* Pengguna\_sistem digunakan untuk menangani halaman pengguna sistem dan beberapa *function* pengelolaan data pengguna sistem. *Class* Pegawai digunakan untuk menangani halaman pegawai dan beberapa *function* pengelolaan data pegawai. *Class* Daftar\_tujuan digunakan untuk menangani halaman daftar tujuan dan beberapa *function* pengelolaan data daftar tujuan. *Class* Armada\_bus digunakan untuk menangani halaman armada bus dan beberapa *function* pengelolaan data armada bus. *Class* CI\_Controller yang akan di *extend* oleh semua *class* yang telah dijelaskan sebelumnya. *Class* CI\_Controller merupakan *class* yang telah ada dalam *framework codeigniter*.





**Gambar 5.7 Class Diagram sebagai Domain Model**

Pada gambar 5.7 merupakan *class diagram model* yang berfungsi untuk melakukan akses terhadap database. Terdapat 7 model yang mempunyai fungsi yang berbeda-beda. Model *M\_armada\_bus* digunakan untuk mengelola data yang terdapat pada tabel armada bus dalam database. Model *M\_pegawai* digunakan untuk mengelola data yang terdapat pada tabel pegawai. Model *M\_jadwal\_bus* digunakan untuk mengambil data dari beberapa tabel lain hingga dapat menghasilkan data berupa jadwal bus. Model *M\_persewaan* digunakan untuk mengelola tabel persewaan. Model *M\_operasional\_bus* digunakan untuk mengelola operasional bus hingga mengambil data untuk menghasilkan data sebagai laporan. Model *M\_pengguna\_sistem* digunakan untuk mengelola tabel pengguna\_sistem. Model *M\_daftar\_tujuan* digunakan untuk mengelola tabel daftar\_tujuan. Model *CI\_Model* merupakan *class* yang telah ada dalam *framework codeigniter* dan di-*extends* oleh class model yang telah dirancang.

### 4.3 Pemodelan Data

Data dimodelkan dengan menggunakan *Physical Data Modelling* yang dapat dilihat pada gambar 5.8. Terdapat 9 tabel diantaranya, tabel persewaan, tabel gaji, tabel jenis\_pekerjaan, tabel daftar\_tujuan, tabel operasional\_bus, tabel pegawai, tabel operasional\_tambahan\_bus, tabel armada\_bus, dan tabel pengguna\_sistem. Penjelasan detail mengenai setiap tabel akan dijelaskan pada sub bab pemodelan data ini.





Nama tabel : armada\_bus

Fungsi : Untuk menyimpan data - data yang berkaitan dengan armada bus.

**Tabel 5.1 Tabel Armada\_bus**

No.	Nama <i>Field</i>	Type	Lebar	Keterangan
1	id_armada_bus	int	11	Id armada bus
2	id_sopir	Int	11	Id pegawai sopir
3	Id_kernet	Int	11	Id pegawai kernet

No.	Nama <i>Field</i>	Tipe	Lebar	Keterangan
4	Nopol	Varchar	7	No polisi
5	Tahun_buat	Year	4	Tahun buat
6	Kapasitas	Int	11	Kapasitas
7	Jenis_bus	Varchar	10	Jenis bus
8	St_delete	Char	1	Status delete 1 = telah dihapus 0 = ada

#### 4.3.2 Tabel Daftar\_tujuan

Nama tabel : daftar\_tujuan

Jumlah *field* : 5

Fungsi : Untuk menyimpan data - data yang berkaitan dengan daftar tujuan.

**Tabel 5.2 Tabel Daftar\_tujuan**

No.	Nama <i>Field</i>	Tipe	Lebar	Keterangan
1	id_daftar_tujuan	Int	11	id daftar tujuan
2	Tujuan	Varchar	20	Nama tujuan
3	Harga	Double		Harga tujuan
4	Keterangan	Varchar	100	Keterangan
5	St_delete	char	1	Status delete 1 = telah dihapus 0 = ada

#### 4.3.3 Tabel Gaji

Nama tabel : gaji

Jumlah *field* : 4

Fungsi : Untuk menyimpan data - data yang berkaitan dengan gaji.

Tabel 5.3 Tabel Gaji

No.	Nama <i>Field</i>	Tipe	Lebar	Keterangan
1	Id_gaji	Int	11	Id gaji
2	No_kuitansi	Int	11	No kuitansi persewaan
3	Id_armada_bus	Int	11	Id armada bus
4	St_penggajian	Char	1	Status penggajian 1 = telah digaji 0 = belum digaji

#### 4.3.4 Tabel Jenis\_pekerjaan

Nama tabel : jenis\_pekerjaan

Jumlah *field* : 2

Fungsi : Untuk menyimpan data - data yang berkaitan dengan jenis pekerjaan.

Tabel 5.4 Tabel Jenis\_pekerjaan

No.	Nama <i>Field</i>	Tipe	Lebar	Keterangan
1	Id_jenis_pekerjaan	Int	11	Id jenis pekerjaan
2	Pekerjaan	Varchar	50	Nama pekerjaan

#### 4.3.5 Tabel Operasioinal\_bus

Nama tabel : operasional\_bus

Jumlah *field* : 14

Fungsi : Untuk menyimpan data - data yang berkaitan dengan operasional bus.

Tabel 5.5 Tabel Operasional\_bus

No.	Nama <i>Field</i>	Tipe	Lebar	Keterangan
1	Id_operasional_bus	Int	11	Id operasional bus
2	Id_armada_bus	Int	11	Id armada bus
3	No_kuitansi	Varchar	11	No kuitansi

No.	Nama <i>Field</i>	Tipe	Lebar	Keterangan
4	Harga	Double		Harga sewa
5	Bayar_dp	Double		Bayar DP
6	Pot_5_persen	Double		Hasil potongan 5 persen dari harga sewa
7	Hasil_95_persen	Double		Hasil harga sewa – hasil potongan 5 persen
8	Fee	Double		Fee
9	Uang_makan	Double		Uang makan
10	Premi	Double		Premi
11	Solar	Double		Solar
12	Lain_lain	Double		Biaya lain-lain
13	Sopir_2	Int	11	Id pegawai sopir cadangan
14	Kernet_2	Int	11	Id pegawai kernet cadangan

#### 4.3.6 Tabel Operasional\_tambahan

Nama tabel : operasional\_tambahan

Jumlah *field* : 4

Fungsi : Untuk menyimpan data - data yang berkaitan dengan operasional tambahan bus.

**Tabel 5.6 Tabel Operasional\_tambahan**

No.	Nama <i>Field</i>	Tipe	Lebar	Keterangan
1	Id_operasional_tambahan	Int	11	Id operasional tambahan
2	Deskripsi	Varchar	15	Deskripsi
3	Biaya	Double		Biaya
4	Tanggal	Date		Tanggal

#### 4.3.7 Tabel Pegawai

Nama tabel : pegawai

Jumlah *field* : 8

Fungsi : Untuk menyimpan data - data yang berkaitan dengan pegawai.

**Tabel 5.7 Tabel Pegawai**

No.	Nama <i>Field</i>	Tipe	Lebar	Keterangan
1	Id_pegawai	Int	11	Id pegawai
2	No_ktp	Varchar	16	No KTP
3	Id_jenis_pekerjaan	Int	11	Id jenis pekerjaan
4	Nama	Varchar	45	Nama pegawai
5	No_hp	Varchar	14	No HP
6	Alamat	Varchar	45	Alamat
7	Tgl_masuk	Date		Tanggal masuk
8	St_delete	Char	1	Status delete 1 = telah dihapus 0 = ada

#### 4.3.8 Tabel Pengguna\_sistem

Nama tabel : pengguna\_sistem

Jumlah *field* : 13

Fungsi : Untuk menyimpan data - data yang berkaitan dengan pengguna sistem.

**Tabel 5.8 Tabel Pengguna\_sistem**

No.	Nama <i>Field</i>	Tipe	Lebar	Keterangan
1	Id_pengguna_sistem	Int	11	Id pengguna sistem
2	Id_pegawai	Int	11	Id pegawai
3	Nama_pengguna	Varchar	10	Nama pengguna
4	Kata_sandi	Varchar	200	Kata sandi
5	St_lihat_armada	Char	1	Status lihat armada

No.	Nama <i>Field</i>	Tipe	Lebar	Keterangan
				1 = bisa 0= tidak bias
6	St_kel_armada	Char	1	Status kelola armada 1 = bisa 0= tidak bisa
7	St_kel_persewaan	Char	1	Status kelola persewaan 1 = bisa 0= tidak bisa
8	St_kel_operasional	Char	1	Status kelola operasional 1 = bisa 0= tidak bisa
9	St_lap_unit	Char	1	Status laporan unit 1 = bisa 0= tidak bisa
10	St_lap_global	Char	1	Status laporan global 1 = bisa 0= tidak bisa
11	St_slip_gaji	Char	1	Status slip gaji 1 = bisa 0= tidak bisa
12	St_kel_pengguna	Char	1	Status kelola pengguna sistem 1 = bisa 0= tidak bisa
13	St_kel_pegawai	Char	1	Status kelola pegawai 1 = bisa



No.	Nama <i>Field</i>	Tipe	Lebar	Keterangan
				0= tidak bisa

#### 4.3.9 Tabel Persewaan

Nama tabel : persewaan

Jumlah *field* : 10

Fungsi : Untuk menyimpan data - data yang berkaitan dengan persewaan.

**Tabel 5.9 Tabel Persewaan**

No.	Nama <i>Field</i>	Tipe	Lebar	Keterangan
1	No_kuitansi	Int	11	No kuitansi
2	Nama_penyewa	Varchar	45	Nama penyewa
3	Jam_berangkat	Time		Jam berangkat
4	Tgl_berangkat	Date		Tanggal berangkat
5	Tgl_pulang	Date		Tanggal pulang
6	Tgl_lunas	Date		Tanggal pembayaran lunas
7	Tujuan	Varchar	11	Tujuan
8	Tempat_penjemputan	Varchar	15	Tempat penjemputan

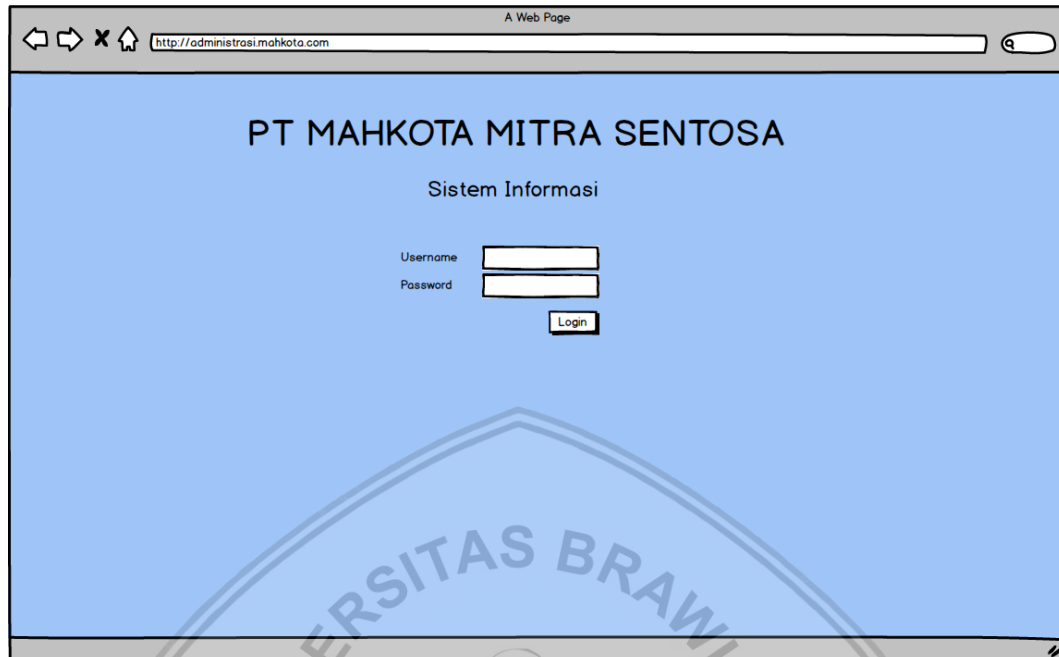
#### 4.4 Antarmuka Pengguna

Pada bagian ini berisi beberapa sketsa tampilan antarmuka pengguna sistem informasi yang akan dikembangkan dalam penelitian ini. Beberapa sketsa antarmuka pengguna sistem di antaranya adalah antarmuka *login*, home, armada bus, jadwal bus, persewaan bus, daftar tujuan, laporan unit, laporan global, gaji, pengguna sistem, pegawai.

##### 4.4.1 Antarmuka *Login*

Antarmuka *login* yang ditunjukkan dalam Gambar 5.9 merupakan tampilan yang akan dilihat oleh pengguna pada saat pengguna akan melakukan autentikasi identitas sebelum dapat mengakses informasi melalui sistem berdasarkan hak aksesnya. Komponen antarmuka *login* di antaranya adalah logo sistem informasi, kolom untuk memasukkan nama atau identitas pengguna,

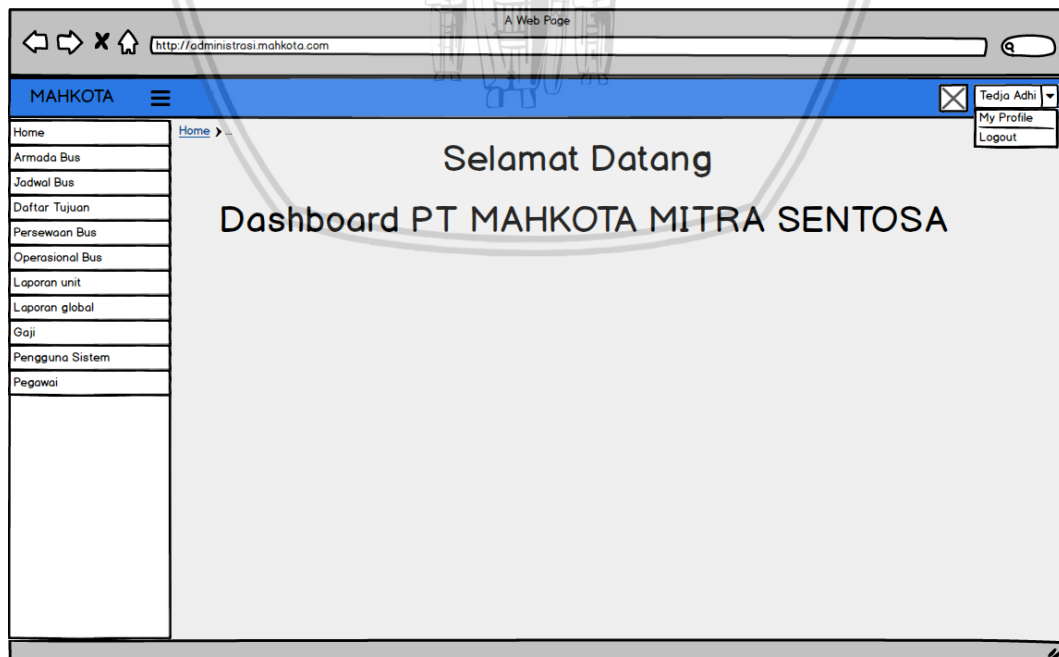
kolom untuk memasukkan kata sandi, dan tombol untuk menjalankan fungsi autentikasi pengguna.



Gambar 5.9 Antarmuka *Login*

#### 4.4.2 Antarmuka *Home*

Antarmuka home yang ditunjukkan dalam Gambar 5.10 merupakan tampilan yang akan dilihat pertama oleh pengguna setelah berhasil masuk ke *dashboard* sistem informasi.



Gambar 5.10 Antarmuka *Home*

#### 4.4.3 Antarmuka Jadwal Bus

Antarmuka jadwal bus yang ditunjukkan dalam Gambar 5.11 merupakan tampilan yang akan dilihat pengguna, pada saat pengguna menggunakan sistem untuk melihat jadwal ketersediaan bus.

The screenshot displays a web browser window with the URL <http://administrasi.mahkota.com>. The page features a blue header with the 'MAHKOTA' logo and a user profile dropdown for 'Tedja Adhi'. A left sidebar contains a menu with items: Home, Armada Bus, Jadwal Bus, Daftar Tujuan, Persewaan Bus, Operasional Bus, Laporan unit, Laporan global, Gaji, Pengguna Sistem, and Pegawai. The main content area is titled 'Pencarian Jadwal Bus' and contains a search form with the following fields: 'Tanggal Berangkat' (departure date) with a calendar icon, 'Tanggal Pulang' (return date) with a calendar icon, and 'Tujuan' (destination). A 'Search' button is located below the 'Tujuan' field. The background of the page features a large, faint watermark of the Universitas Brawijaya logo.

**Gambar 5.11 Antarmuka Jadwal Bus**

Antarmuka jadwal bus yang ditunjukkan dalam Gambar 5.12 merupakan hasil iterasi pertama pada fase elaboration. Pada hasil iterasi ini terdapat perbedaan tampilan dari sebelumnya yang di mana konten tabel terdapat di berbeda halaman, namun dari hasil iterasi konten tabel terdapat dalam halaman yang sama dengan *form* pencarian.

MAHKOTA

Home > Jadwal Bus

### Pencarian Jadwal Bus

Tanggal Berangkat:  /  /

Tanggal Pulang:  /  /

Tujuan:

Biaya dengan tujuan **Bali**  
RP 10.000.000,-

No Polisi	Tahun	Kapasitas/v	Jenis Bus
AG 1 M	2016	60	Besar
AG 2 M	2016	60	Besar
AG 3 M	2016	60	Besar

Gambar 5.12 Antarmuka Jadwal Bus

#### 4.4.4 Antarmuka Persewaan Bus

Antarmuka persewaan bus yang ditunjukkan dalam Gambar 5.13 merupakan tampilan pertama yang akan dilihat pengguna, pada saat pengguna akan menggunakan sistem untuk melakukan kelola persewaan bus.

MAHKOTA

Home > Persewaan Bus

### Data Persewaan Bus Mahkota

No Kwitansi	Nama Penyewa	Tgl Berangkat	Tgl Pulang	Tgl Lunas	Tujuan	Action
2018-01-10	Mita Sugiarti	2018-01-10	2018-01-15	2018-01-10	Bali	<input type="button" value="View"/> <input type="button" value="Edit"/>
2018-01-11	Vivi Sugiarti	2018-01-10	2018-01-15	2018-01-10	Bali	<input type="button" value="View"/> <input type="button" value="Edit"/>
2018-01-12	Anti Sugiarti	2018-01-10	2018-01-15	2018-01-10	Bali	<input type="button" value="View"/> <input type="button" value="Edit"/>
2018-01-13	Seni Sugiarti	2018-01-10	2018-01-15	2018-01-10	Bali	<input type="button" value="View"/> <input type="button" value="Edit"/>
2018-01-13	Ani Sugiarti	2018-01-10	2018-01-15	2018-01-10	Bali	<input type="button" value="View"/> <input type="button" value="Edit"/>

Gambar 5.13 Antarmuka Persewaan Bus

#### 4.4.5 Antarmuka Persewaan Bus : Menambah Persewaan

Antarmuka menambah data persewaan bus yang ditunjukkan dalam Gambar 5.14 merupakan tampilan yang akan dilihat pengguna, pada saat pengguna menggunakan system untuk menambah data persewaan persewaan bus.

Gambar 5.14 Antarmuka Persewaan Bus : Menambah Persewaan

#### 4.4.6 Antarmuka Persewaan Bus : Menyunting Persewaan

Antarmuka menyunting data persewaan bus yang ditunjukkan dalam Gambar 5.15 merupakan tampilan yang akan dilihat pengguna, pada saat pengguna menggunakan sistem untuk menyunting data persewaan persewaan bus.

Gambar 5.15 Antarmuka Persewaan Bus : Menyunting Persewaan

#### 4.4.7 Antarmuka Laporan Unit

Antarmuka laporan unit dari persewaan bus yang ditunjukkan dalam Gambar 5.16 merupakan tampilan yang akan dilihat pengguna, pada saat pengguna menggunakan sistem untuk melihat laporan unit atau untuk mencetak laporan persewaan bus.

The screenshot shows a web browser window with the URL <http://administrasi.mahkota.com>. The page title is 'MAHKOTA'. The sidebar menu includes: Home, Armada Bus, Jadwal Bus, Daftar Tujuan, Persewaan Bus, Operasional Bus, Laporan unit, Laporan global, Gaji, Pengguna Sistem, and Pegawai. The main content area is titled 'Laporan Unit' and contains the following form elements:

- Periode:  /
- No Polisi:
- Show button

Gambar 5.16 Antarmuka Laporan Unit

Antarmuka laporan unit dari persewaan bus yang ditunjukkan dalam Gambar 5.17 merupakan hasil iterasi pertama pada fase elaboration. Pada hasil iterasi ini terdapat perbedaan tampilan dari sebelumnya yang di mana konten tabel terdapat di berbeda halaman, namun dari hasil iterasi konten tabel terdapat dalam halaman yang sama dengan *form* pencarian.

The screenshot shows the same web browser window as Gambar 5.16, but with additional features:
 

- Print, Excel, and Pdf buttons in the top right corner.
- A search bar with a magnifying glass icon and the text 'search'.
- A 'Show' dropdown menu set to '5'.
- A table with the following data:

Tgl	Penyewa	Tujuan	Jam	Tgl Berangkat	DP	Lunas	Penjemputan	Harga	Pot 5%
1	Bowo	Bali	07:00	1-2	0	16-01-2018	Kediri	10.000.000	500.000
2	~	~	~	~	~	~	~	~	~
3									
4									
5									
6	Bowo	Bali	07:00	6-8	0	16-01-2018	Kediri	10.000.000	500.000
7	~	~	~	~	~	~	~	~	~
8	~	~	~	~	~	~	~	~	~

Gambar 5.17 Antarmuka Laporan Unit (hasil iterasi)



#### 4.4.8 Antarmuka Laporan Global

Antarmuka laporan global dari persewaan bus yang ditunjukkan dalam Gambar 5.18 merupakan tampilan yang akan dilihat pengguna, pada saat pengguna menggunakan sistem untuk melihat laporan unit atau untuk mencetak laporan persewaan bus.

The screenshot shows a web browser window with the URL <http://administrasi.mahkota.com>. The page title is 'A Web Page'. The sidebar menu on the left lists various system functions. The main content area is titled 'Laporan Global' and contains a 'Periode' input field and a 'Show' button.

Gambar 5.18 Antarmuka Laporan Global

Antarmuka laporan global dari persewaan bus yang ditunjukkan dalam Gambar 5.19 merupakan hasil iterasi pertama pada fase elaboration. Pada hasil iterasi ini terdapat perbedaan tampilan dari sebelumnya yang di mana konten tabel terdapat di berbeda halaman, namun dari hasil iterasi konten tabel terdapat dalam halaman yang sama dengan *form* pencarian.

The screenshot shows the same web browser window as Gambar 5.18, but with additional features. Below the 'Show' button, there is a 'Print', 'Excel', and 'Pdf' button group, and a search bar. The table below shows the data for the bus units.

No	NoPol	Nama	Harga	Fee+GtBangku	Premi+Uj Makan	Solar	Sisa	Pemasukkan Lain	Jumlah
1	AG 1 M	Agus A	10.000.000	250.000	1425.000	1500.000	6.825.000	0	
2	AG 2 M	Agus B	10.000.000	250.000	1425.000	1500.000	6.825.000	0	
3	AG 3 M	Agus C	10.000.000	250.000	1425.000	1500.000	6.825.000	0	
4	AG 4 M	Agus D	10.000.000	250.000	1425.000	1500.000	6.825.000	0	
5	AG 5 M	Agus E	10.000.000	250.000	1425.000	1500.000	6.825.000	0	
6	AG 6 M	Agus F	10.000.000	250.000	1425.000	1500.000	6.825.000	0	
7	AG 7 M	Agus G	10.000.000	250.000	1425.000	1500.000	6.825.000	0	
8	AG 8 M	Agus H	10.000.000	250.000	1425.000	1500.000	6.825.000	0	
9	AG 9 M	Agus I	10.000.000	250.000	1425.000	1500.000	6.825.000	0	
10	AG 10 M	Agus J	10.000.000	250.000	1425.000	1500.000	6.825.000	0	

Gambar 5.19 Antarmuka Laporan Global (hasil iterasi)

#### 4.4.9 Antarmuka Gaji

Antarmuka gaji sopir dan kernet yang ditunjukkan dalam Gambar 5.20 merupakan tampilan yang akan dilihat pengguna, pada saat pengguna menggunakan system untuk mencari data gaji sopir dan kernet dari data persewaan bus.

Gambar 5.20 Antarmuka Gaji

#### 4.4.10 Antarmuka Gaji : Cetak Slip Gaji

Antarmuka slip gaji yang ditunjukkan dalam Gambar 5.21 merupakan tampilan yang akan dilihat pengguna, pada saat pengguna menggunakan sistem untuk melihat dan mencetak slip gaji.

Gambar 5.21 Antarmuka Gaji : Cetak Slip Gaji

## 4.5 Perancangan Algoritme

Perancangan algoritme merupakan tahap yang dilakukan untuk menentukan dan menyusun sekumpulan operasi logika sebagai konsep pembangunan fungsi dari sistem informasi yang akan dikembangkan. Algoritme yang telah didokumentasikan akan menjadi panduan untuk pembangunan sistem informasi dalam tahap implementasi.

### 4.5.1 Algoritme fungsi displayJadwalBus()

Algoritme fungsi displayJadwalBus() merupakan algoritme untuk mengambil data dari database dan menampilkan halaman jadwal bus. Tabel 5.10 menjelaskan bagaimana urutan dari proses sistem menampilkan halaman jadwal bus.

**Tabel 5.10 Algoritme fungsi displayJadwalBus()**

No	Algoritme
1	Mulai
2	Menampung judul halaman ke dalam variable data array
3	Menampung data daftar tujuan dari database ke dalam variable data array
4	Memanggil library template dengan beberapa argument untuk menampilkan halaman jadwal bus
5	Selesai

### 4.5.2 Algoritme fungsi getJadwalBus()

Algoritme fungsi getJadwalBus() merupakan algoritme untuk mengambil data dari database sesuai dengan data yang telah dimasukkan oleh pengguna sistem dan menampilkan data tersebut ke halaman jadwal bus. Tabel 5.11 menjelaskan bagaimana urutan dari proses sistem menampilkan data dari database ke halaman jadwal bus.

**Tabel 5.11 Algoritme fungsi getJadwalBus()**

No	Algoritme
1	Mulai
2	Menampung masukkan tgl berangkat ke variable
3	Menampung masukkan tgl pulang ke variable
4	Menampung hasil explode data tgl_berangkat ke variable
5	Menampung hasil explode data tgl_pulang ke variable
6	Menampung data tgl_berangkat dan tgl_pulang ke variable array
7	Menampung data dari database ke variable array res
8	Jika data dari database bisa dihitung
9	Menampung nilai false
10	Menampung nilai false
11	Menampung jumlah data
12	Lainnya

No	Algoritme
13	Menampung nilai true
14	Menampung nilai true
15	Menampung pesan
16	Mengembalikan nilai variable array res
17	Selesai

#### 4.5.3 Algoritme fungsi displayPersewaanBus()

Algoritme fungsi displayPersewaanBus() merupakan algoritme untuk menampilkan halaman persewaan bus. Tabel 5.12 menjelaskan bagaimana urutan dari proses sistem menampilkan halaman persewaan bus.

**Tabel 5.12 Algoritme fungsi displayPersewaanBus()**

No	Algoritme
1	Mulai
2	Menampung judul halaman ke variable array
3	Memanggil library template dengan beberapa argument
4	untuk menampilkan halaman persewaan bus
5	Selesai

#### 4.5.4 Algoritme fungsi addPersewaanBus()

Algoritme fungsi addPersewaanBus() merupakan algoritme untuk menerima hasil masukkan pengguna sistem dan menyimpannya ke dalam *database*. Tabel 5.13 menjelaskan bagaimana urutan dari proses sistem menambah data persewaan bus.

**Tabel 5.13 Algoritme fungsi addPersewaanBus()**

No	Algoritme
1	Mulai
2	Manampung hasil explode dari masukkan tgl berangkat
3	Manampung hasil explode dari masukkan tgl pulang
4	Manampung hasil explode dari masukkan tgl lunas
5	Menampung semua masukkan ke variable array
6	Menampung hasil query input database(tabel persewaan)
7	Jika hasil query input database(tabel persewaan)
8	true
9	Menampung hasil query input database(tabel operasional bus)
10	Menampung hasil query input database(tabel gaji)
11	Manampung nilai false ke variable
12	Lainnya
13	Manampung nilai false ke variable
14	Mengembalikan nilai variable
15	Selesai

#### 4.5.5 Algoritme fungsi updatePersewaanBus()

Algoritme fungsi updatePersewaanBus() merupakan algoritme untuk menerima hasil masukkan pengguna sistem dan menyimpannya ke dalam *database*. Tabel 5.14 menjelaskan bagaimana urutan dari proses sistem menyunting data persewaan bus.

**Tabel 5.14 Algoritme fungsi updatePersewaanBus()**

No	Algoritme
1	Mulai
2	Manampung hasil explode dari masukkan tgl berangkat
3	Manampung hasil explode dari masukkan tgl pulang
4	Manampung hasil explode dari masukkan tgl lunas
5	Menampung semua masukkan ke variable array
6	Menampung hasil query update database (tabel persewaan)
7	Menampung hasil query update database (tabel operasional)
8	Mengembalikan nilai variable
9	Selesai

#### 4.5.6 Algoritme fungsi getLaporanUnit()

Algoritme fungsi getLaporanUnit() merupakan algoritme untuk menampilkan data dari *database* sesuai dengan masukkan pengguna sistem ke halaman laporan unit. Tabel 5.15 menjelaskan bagaimana urutan dari proses sistem menampilkan data laporan unit pada halaman laporn unit.

**Tabel 5.15 Algoritme fungsi getLaporanUnit()**

No	Algoritme
1	Mulai
2	Menampung masukkan periode
3	Manampung hasil explode periode
4	Manampung semua masukkan ke variable array
5	Menampung hasil query
6	Jika hasil query dapat dihitung
7	Manampung nilai false ke variable array
8	Menampung jumlah data hasil query ke variable array
9	Lainnya
10	Manampung nilai true ke variable array
11	Manampung pesan ke variable array
12	Mengembalikan nilai variable array
13	Selesai

#### 4.5.7 Algoritme fungsi getLaporanGlobal()

Algoritme fungsi getLaporanGlobal() merupakan algoritme untuk menampilkan data dari *database* sesuai dengan masukkan pengguna sistem ke

halaman laporan global. Tabel 5.16 menjelaskan bagaimana urutan dari proses sistem menampilkan data laporan global pada halaman laporn global.

**Tabel 5.16 Algoritme fungsi getLaporanGlobal()**

No	Algoritme
1	Mulai
2	Menampung masukkan periode
3	Manampung hasil explode periode
4	Manampung semua masukkan ke variable array
5	Menampung hasil query
6	Jika hasil query dapat dihitung
7	Manampung nilai false ke variable array
8	Menampung jumlah data hasil query ke variable array
9	Lainnya
10	Manampung nilai true ke variable array
11	Manampung pesan ke variable array
12	Mengembalikan nilai variable array
13	Selesai

#### 4.5.8 Algoritme fungsi getStatusGaji()

Algoritme fungsi getStatusGaji() merupakan algoritme untuk memperoleh data status gaji dari *database*. Tabel 5.17 menjelaskan bagaimana urutan dari proses sistem memperoleh data status gaji.

**Tabel 5.17 Algoritme fungsi getStatusGaji()**

No	Algoritme
1	Mulai
2	Menampung semua masukkan ke variable array
3	Menampung hasil query ke variable array
4	Jika data hasil query dapat dihitung
5	Menampung nilai false ke variable array
6	Menampung jumlah data hasil query ke variable array
7	Lainnya
8	Menampung nilai true ke variable array
9	Menampung pesan ke variable array
10	Mengembalikan linai variable array
11	Selesai

#### 4.5.9 Algoritme fungsi displayGajiPrint()

Algoritme fungsi displayGajiPrint() merupakan algoritme untuk menampilkan data gaji dari *database* sesuai dengan masukkan pengguna sistem ke halaman gaji print. Tabel 5.18 menjelaskan bagaimana urutan dari proses sistem menampilkan data gaji pada halaman gaji print yang digunakan untuk mencetak slip gaji.



Tabel 5.18 Algoritme fungsi displayGajiPrint()

No	Algoritme
1	Mulai
2	Manampung judul halaman ke variable array
3	Menampung hasil query ke variable array
4	Menampung hasil query ke variable array
5	Menampung hasil perhitungan pembulatan gaji sopir utama
6	Manampung hasil operasi string
7	Menampung hasil akhir gaji sopir utama
8	Menampung nilai 0 untuk sopir cadangan
9	Manampung hasil operasi string
10	Menampung hasil akhir gaji sopir cadangan
11	Menampung hasil perhitungan pembulatan gaji kernet utama
12	Manampung hasil operasi string
13	Menampung hasil akhir gaji kernet utama
14	Menampung nilai 0 untuk kernet cadangan
15	Manampung hasil operasi string
16	Menampung hasil akhir gaji kernet cadangan
17	Jika data sopir cadangan kosong
18	Menampung nilai 0 sopir cadangan
19	Lain jika kernet cadangan kosong
20	Manampung nilai 0 kernet cadangan
21	Lain jika sopir cadangan tidak kosong
22	Menampung hasil akhir gaji sopir utama
23	Menampung nilai 0 untuk sopir cadangan
24	Manampung hasil operasi string
25	Menampung hasil akhir gaji sopir cadangan
26	Lain jika kernet cadangan tidak kosong
27	Menampung hasil akhir gaji kernet utama
28	Menampung nilai 0 untuk kernet cadangan
29	Manampung hasil operasi string
30	Menampung hasil akhir gaji kernet cadangan
31	Memanggil library template dengan beberapa argument
32	untuk menampilkan halaman slip gaji untuk dicetak
33	Selesai

## BAB 6 IMPLEMENTASI

### 6.1 Spesifikasi Lingkungan Implementasi

Subbab ini menjelaskan spesifikasi perangkat keras dan perangkat lunak yang digunakan selama proses pembangunan sistem informasi. Perangkat keras yang digunakan pada proses pembangunan sistem informasi adalah sebuah unit laptop dengan spesifikasi yang dijelaskan pada Tabel 6.1. Selain itu terdapat informasi spesifikasi perangkat lunak yang digunakan untuk pengembangan ditunjukkan pada Tabel 6.2.

**Tabel 6.1 Spesifikasi Perangkat Keras**

<b>Unit Komputasi</b>	Laptop ACER E5-471G
<b>CPU</b>	Core i3-4030U 1.90 Ghz
<b>Kapasitas RAM</b>	10 GB
<b>Kapasitas Penyimpanan</b>	500 GB
<b>Kartu Grafis</b>	Intel(R) HD Graphics Family
<b>Resolusi Layar</b>	1366 x 768 pixels

**Tabel 6.2 Spesifikasi Perangkat Lunak**

<b>Sistem Operasi</b>	Microsoft Windows 10 Pro
<b>Web Server</b>	Apache 2.4.33
<b>DBMS</b>	MariaDB 10.1.31
<b>Bahasa Pemrograman</b>	PHP 7.2.4
<b>Editor Kode Program</b>	Microsoft Visual Code
<b>Peramban</b>	Google Chrome Version 66.0.3359.181 (Official Build) (64-bit)
<b>Perangkat Lunak atau Bahasa Pemrograman Pendukung</b>	Javascript, HTML

## 6.2 Implementasi Algoritme

Subbab implementasi algoritme berisi beberapa kode program yang dibuat berdasarkan perancangan algoritme yang telah dibuat pada tahap perancangan. Hasil implementasi algoritme menggunakan bahasa PHP.

### 6.2.1 Algoritme fungsi displayJadwalBus()

Implementasi algoritme dalam bentuk kode program yang terdapat pada Tabel 6.3 merupakan implementasi fungsi untuk mengambil data dari *database* dan menampilkan halaman jadwal bus. Implementasi algoritme dilakukan berdasarkan hasil rancangan algoritme fungsi displayJadwalBus().

**Tabel 6.3 Implementasi Algoritme fungsi displayJadwalBus()**

No.	Kode Program
1	public function displayJadwalBus() {
2	\$data['title'] = 'Jadwal Bus   PT MAHKOTA MITRA SENTOSA';
3	\$data['tujuan'] = \$this->m_daftar_tujuan->getDaftarTujuan();
4	\$this->template->display('content/jadwal_bus', \$data);
5	}

### 6.2.2 Algoritme fungsi getJadwalBus()

Implementasi algoritme dalam bentuk kode program yang terdapat pada Tabel 6.4 merupakan implementasi fungsi untuk mengambil data dari *database* sesuai dengan data yang telah dimasukkan oleh pengguna sistem dan menampilkan data tersebut ke halaman jadwal bus. Implementasi algoritme dilakukan berdasarkan hasil rancangan algoritme fungsi getJadwalBus().

**Tabel 6.4 Implementasi Algoritme fungsi getJadwalBus()**

No.	Kode Program
1	public function getJadwalBus() {
2	\$ttl_berangkat = (!empty(\$this->input->post('tglberangkat')) ? \$this->input->post('tglberangkat') : NULL);
3	\$ttl_pulang = (!empty(\$this->input->post('tglpulang')) ? \$this->input->post('tglpulang') : NULL);
4	\$exp_tgl_berangkat = explode("-", \$ttl_berangkat);
5	\$exp_tgl_pulang = explode("-", \$ttl_pulang);
6	\$params = array(
7	"tgl_berangkat" =>
8	\$exp_tgl_berangkat[2]."-".\$exp_tgl_berangkat[1]."-". \$exp_tgl_berangkat[0],

No.	Kode Program
9	<code>"tgl_pulang" =&gt; \$exp_tgl_pulang[2]."-</code>
10	<code>".\$exp_tgl_pulang[1]."-". \$exp_tgl_pulang[0]</code>
	<code>);</code>
	<code>\$res["Data"] = \$this-&gt;m_jadwal_bus-</code>
11	<code>&gt;getJadwalBus(\$params);</code>
12	<code>if(count(\$res["Data"])){</code>
13	<code>\$res["IsError"] = FALSE;</code>
14	<code>\$res["IsNull"] = FALSE;</code>
	<code>\$res["Messages"] = count(\$res["Data"])."</code>
15	<code>Data Ditemukan";</code>
16	<code>} else {</code>
17	<code>\$res["IsError"] = TRUE;</code>
18	<code>\$res["IsNull"] = TRUE;</code>
19	<code>\$res["Messages"] = "Data Tidak Ditemukan!";</code>
20	<code>}</code>
21	<code>echo json_encode(\$res);</code>
	<code>}</code>

### 6.2.3 Algoritme fungsi displayPersewaanBus()

Implementasi algoritme dalam bentuk kode program yang terdapat pada Tabel 6.5 merupakan implementasi fungsi untuk menampilkan halaman persewaan bus. Implementasi algoritme dilakukan berdasarkan hasil rancangan algoritme fungsi displayPersewaanBus().

**Tabel 6.5 Implementasi Algoritme fungsi displayPersewaanBus()**

No.	Kode Program
1	<code>public function displayPersewaanBus(){</code>
2	<code>\$data['title']='Persewaan Bus   PT MAHKOTA</code>
	<code>MITRA SENTOSA';</code>
3	<code>\$this-&gt;template-</code>
	<code>&gt;display('content/persewaan_bus',\$data);</code>
4	<code>}</code>

### 6.2.4 Algoritme fungsi addPersewaanBus()

Implementasi algoritme dalam bentuk kode program yang terdapat pada Tabel 6.6 merupakan implementasi fungsi untuk menerima hasil masukkan pengguna sistem dan menyimpannya ke dalam *database*. Implementasi algoritme dilakukan berdasarkan hasil rancangan algoritme fungsi addPersewaanBus().

**Tabel 6.6 Implementasi Algoritme fungsi addPersewaanBus()**

No.	Kode Program
1	<code>public function addPersewaanBus(){</code>
2	

No.	Kode Program
3	<code>\$tgl_berangkat = explode("-", \$this-&gt;input-&gt;post('tgl_berangkat'));</code>
4	<code>\$tgl_pulang = explode("-", \$this-&gt;input-&gt;post('tgl_pulang'));</code>
5	<code>\$tgl_lunas = explode("-", \$this-&gt;input-&gt;post('tgl_lunas'));</code>
6	<code>\$params = array( "no_kuitansi"=&gt;\$this-&gt;input-&gt;post('kuitansi'), "nama penyewa"=&gt;\$this-&gt;input-&gt;post('nama penyewa'), "jam berangkat"=&gt;\$this-&gt;input-&gt;post('jam berangkat'), "tpt penjemputan"=&gt;\$this-&gt;input-&gt;post('tpt penjemputan'), "tgl berangkat" =&gt; \$tgl_berangkat[2]."-". \$tgl_berangkat[1]."-". \$tgl_berangkat[0], "tgl pulang" =&gt; \$tgl_pulang[2]."-". \$tgl_pulang[1]."-". \$tgl_pulang[0], "tgl lunas" =&gt; \$tgl_lunas[2]."-". \$tgl_lunas[1]."-". \$tgl_lunas[0], "tujuan"=&gt;\$this-&gt;input-&gt;post('tujuan'), "info armada"=&gt;\$this-&gt;input-&gt;post('data-repeat') );</code>
17	<code>\$query['tambah-persewaan'] = \$this-&gt;m_persewaan-&gt;addPersewaan(\$params);</code>
18	<code>if(\$query['tambah-persewaan']==true){ \$query['tambah-operasional'] = \$this-&gt;m_operasional_bus-&gt;addOperasionalBus(\$params); \$query['tambah-gaji'] = \$this-&gt;m_gaji-&gt;addStGaji(\$params);</code>
21	<code>\$IsError = false;</code>
22	<code>}else{</code>
23	<code>\$IsError = true;</code>
24	<code>}</code>
25	<code>echo json_encode(\$IsError); }</code>

### 6.2.5 Algoritme fungsi updatePersewaanBus()

Implementasi algoritme dalam bentuk kode program yang terdapat pada Tabel 6.7 merupakan implementasi fungsi untuk menerima hasil masukan pengguna sistem dan menyimpannya ke dalam *database*. Implementasi algoritme dilakukan berdasarkan hasil rancangan algoritme fungsi updatePersewaanBus().

**Tabel 6.7 Implementasi Algoritme fungsi updatePersewaanBus()**

No.	Kode Program
1	public function updatePersewaanBus() {
2	\$ttl_berangkat = explode("-", \$this->input->post('ttl_berangkat'));
3	\$ttl_pulang = explode("-", \$this->input->post('ttl_pulang'));
4	\$ttl_lunas = explode("-", \$this->input->post('ttl_lunas'));
5	\$params = array(
6	"no_kuitansi"=>\$this->input->post('kuitansi'),
7	"nama_penyewa"=>\$this->input->post('nama_penyewa'),
8	"jam_berangkat"=>\$this->input->post('jam_berangkat'),
9	"tpt_penjemputan"=>\$this->input->post('tpt_penjemputan'),
10	"ttl_berangkat" => \$ttl_berangkat[2]."-".
11	"\$ttl_berangkat[1]."-".\$ttl_berangkat[0],
12	"ttl_pulang" => \$ttl_pulang[2]."-".
13	"\$ttl_pulang[1]."-".\$ttl_pulang[0],
14	"ttl_lunas" => \$ttl_lunas[2]."-".
15	"\$ttl_lunas[1]."-".\$ttl_lunas[0],
16	"tujuan"=>\$this->input->post('tujuan'),
17	"info_armada"=>\$this->input->post('data-repeat')
18	);
19	\$query['update-persewaan'] = \$this->m_persewaan->updatePersewaan(\$params);
	\$query['update-operasional'] = \$this->m_operasional_bus->updateOperasionalBus(\$params);
	echo json_encode(\$query);
	}

### 6.2.6 Algoritme fungsi getLaporanUnit()

Implementasi algoritme dalam bentuk kode program yang terdapat pada Tabel 6.8 merupakan implementasi fungsi untuk menampilkan data dari *database* sesuai dengan masukkan pengguna sistem ke halaman laporan unit. Implementasi algoritme dilakukan berdasarkan hasil rancangan algoritme fungsi getLaporanUnit().

**Tabel 6.8 Implementasi Algoritme fungsi getLaporanUnit()**

No.	Kode Program
1	public function getLaporanUnit() {
2	\$periode = (!empty(\$this->input->post('periode')) ? \$this->input->post('periode') : NULL);



No.	Kode Program
3	<code>\$exp_periode = explode("/", \$periode);</code>
4	<code>\$params = array(</code>
5	<code>    "bulan" =&gt; \$exp_periode[0],</code>
6	<code>    "tahun" =&gt; \$exp_periode[1],</code>
7	<code>    "id_armada_bus" =&gt; (!empty(\$this-&gt;input-&gt;post('nopol'))?\$this-&gt;input-&gt;post('nopol'):NULL)</code>
8	<code>);</code>
9	<code>\$res["Data"] = \$this-&gt;m_operasional_bus-&gt;getLaporanUnit(\$params);</code>
10	<code>    if(count(\$res["Data"])){</code>
11	<code>        \$res["IsError"] = FALSE;</code>
12	<code>        \$res["Messages"] = count(\$res["Data"])."</code>
13	<code>        Data Found";</code>
14	<code>    } else {</code>
15	<code>        \$res["IsError"] = TRUE;</code>
16	<code>        \$res["Messages"] = "Data Not Found";</code>
17	<code>    }</code>
18	<code>    echo json_encode(\$res);</code>

### 6.2.7 Algoritme fungsi getLaporanGlobal()

Implementasi algoritme dalam bentuk kode program yang terdapat pada Tabel 6.9 merupakan implementasi fungsi untuk menampilkan data dari *database* sesuai dengan masukkan pengguna sistem ke halaman laporan global. Implementasi algoritme dilakukan berdasarkan hasil rancangan algoritme fungsi getLaporanGlobal().

**Tabel 6.9 Implementasi Algoritme fungsi getLaporanGlobal()**

No.	Kode Program
1	<code>public function getLaporanGlobal(){</code>
2	<code>    \$periode = (!empty(\$this-&gt;input-&gt;post('periode'))?\$this-&gt;input-&gt;post('periode'):NULL);</code>
3	<code>    \$exp_periode = explode("/", \$periode);</code>
4	<code>    \$params = array(</code>
5	<code>        "bulan" =&gt; \$exp_periode[0],</code>
6	<code>        "tahun" =&gt; \$exp_periode[1]</code>
7	<code>    );</code>
8	<code>    \$res["Data"] = \$this-&gt;m_operasional_bus-&gt;getLaporanGlobal(\$params);</code>
9	<code>    if(count(\$res["Data"])){</code>
10	<code>        \$res["IsError"] = FALSE;</code>
11	<code>        \$res["Messages"] = count(\$res["Data"])."</code>
12	<code>        Data Found";</code>
13	<code>    } else {</code>
14	<code>        \$res["IsError"] = TRUE;</code>
15	<code>        \$res["Messages"] = "Data Not Found";</code>
16	<code>    }</code>

No.	Kode Program
16	<code>echo json_encode(\$res);</code>
17	<code>}</code>

### 6.2.8 Algoritme fungsi getStatusGaji()

Implementasi algoritme dalam bentuk kode program yang terdapat pada Tabel 6.10 merupakan implementasi fungsi untuk memperoleh data status gaji dari *database*. Implementasi algoritme dilakukan berdasarkan hasil rancangan algoritme fungsi getStatusGaji().

**Tabel 6.10 Implementasi Algoritme fungsi getStatusGaji()**

No.	Kode Program
1	<code>public function getStatusGaji(){</code>
2	<code>    \$no_kuitansi = \$this-&gt;input-</code>
3	<code>&gt;post('no_kuitansi');</code>
4	<code>    \$id_armada_bus = \$this-&gt;input-</code>
5	<code>&gt;post('id_armada_bus');</code>
6	<code>    \$res["Data"] = \$this-&gt;m_operasional_bus-</code>
7	<code>&gt;getOperasionalBus(\$no_kuitansi,\$id_armada_bus);</code>
8	<code>    if(count(\$res["Data"])){</code>
9	<code>        \$res["IsNull"] = FALSE;</code>
10	<code>        \$res["Messages"] = count(\$res["Data"])."</code>
11	<code>        Data Found";</code>
12	<code>    } else {</code>
13	<code>        \$res["IsNull"] = TRUE;</code>
14	<code>        \$res["Messages"] = "Data Not Found";</code>
15	<code>    }</code>
16	<code>    echo json_encode(\$res);</code>
17	<code>}</code>

### 6.2.9 Algoritme fungsi displayGajiPrint()

Implementasi algoritme dalam bentuk kode program yang terdapat pada Tabel 6.11 merupakan implementasi fungsi untuk menampilkan data gaji dari *database* sesuai dengan masukkan pengguna sistem ke halaman gaji print. Implementasi algoritme dilakukan berdasarkan hasil rancangan algoritme fungsi displayGajiPrint().

**Tabel 6.11 Implementasi Algoritme fungsi displayGajiPrint()**

No.	Kode Program
1	<code>Public function</code>
2	<code>displayGajiPrint(\$no_kuitansi,\$id_armada_bus){</code>
3	<code>    \$data['title'] = 'GAJI   PT MAHKOTA MITRA</code>
4	<code>    SENTOSA';</code>
5	<code>    \$data['gaji'] = \$this-&gt;m_operasional_bus-</code>
6	<code>&gt;getGaji(\$no_kuitansi,\$id_armada_bus);</code>

No.	Kode Program
5	<code>\$data['st_gaji'] = \$this-&gt;m_gaji-</code>
	<code>&gt;getStatusGaji(\$no_kuitansi,\$id_armada_bus);</code>
6	<code>\$data['sopir_utama'] =</code>
	<code>floor((\$data['gaji'][0]['premi'])*10/15);</code>
7	<code>\$ratusan = substr(\$data['sopir_utama'],-3);</code>
8	<code>\$data['sopir_utama'] = \$data['sopir_utama']-</code>
	<code>(\$ratusan);</code>
9	<code>\$data['sopir_cadangan'] = "0";</code>
10	<code>\$ratusan = substr(\$data['sopir_cadangan'],-3);</code>
11	<code>\$data['sopir_cadangan'] =</code>
	<code>\$data['sopir_cadangan']-(\$ratusan);</code>
12	<code>\$data['kernet_utama'] =</code>
	<code>floor((\$data['gaji'][0]['premi'])*5/15);</code>
13	<code>\$ratusan = substr(\$data['kernet_utama'],-3);</code>
14	<code>\$data['kernet_utama'] = \$data['kernet_utama']-</code>
	<code>(\$ratusan);</code>
15	<code>\$data['kernet_cadangan'] = "0";</code>
16	<code>\$ratusan = substr(\$data['kernet_cadangan'],-</code>
	<code>3);</code>
17	<code>\$data['kernet_cadangan'] =</code>
	<code>\$data['kernet_cadangan']-(\$ratusan);</code>
18	<code>if(empty(\$data['gaji'][0]['sopir_2'])) {</code>
19	<code>\$data['sopir_cadangan'] = "0";</code>
20	<code>}elseif(empty(\$data['gaji'][0]['kernet_2'])) {</code>
21	<code>\$data['kernet_cadangan'] = "0";</code>
22	<code>}elseif(!empty(\$data['gaji'][0]['sopir_2'])) {</code>
23	<code>\$gaji_sopir =</code>
	<code>floor((\$data['gaji'][0]['premi'])*14/19)/2;</code>
24	<code>\$data['sopir_utama'] = \$gaji_sopir;</code>
25	<code>\$data['sopir_cadangan'] = \$gaji_sopir;</code>
26	<code>}elseif(!empty(\$data['gaji'][0]['kernet_2'])) {</code>
27	<code>\$gaji_kernet =</code>
	<code>floor((\$data['gaji'][0]['premi'])*5/19);</code>
28	<code>\$data['kernet_utama'] = "0";</code>
29	<code>\$data['kernet_cadangan'] = \$gaji_kernet;</code>
30	<code>}</code>
31	<code>\$this-&gt;template-</code>
	<code>&gt;display('content/gaji_print',\$data);</code>
32	<code>}</code>

### 6.3 Implementasi Antarmuka Pengguna

Subbab implementasi antarmuka pengguna menjelaskan beberapa hasil implementasi dalam gambar antarmuka pengguna sistem informasi. Subbab ini hanya menyertakan beberapa hasil implementasi antarmuka pengguna di antaranya adalah antarmuka *login*, antarmuka home, antarmuka jadwal bus, antarmuka persewaan bus, antarmuka menambah persewaan, antarmuka menyunting persewaan, antarmuka laporan unit, antarmuka laporan global, antarmuka gaji, antarmuka slip gaji. Implementasi antarmuka pengguna

dilakukan berdasarkan hasil perancangan antarmuka pengguna yang telah dilakukan dalam penelitian ini.

### 6.3.1 Antarmuka *Login*

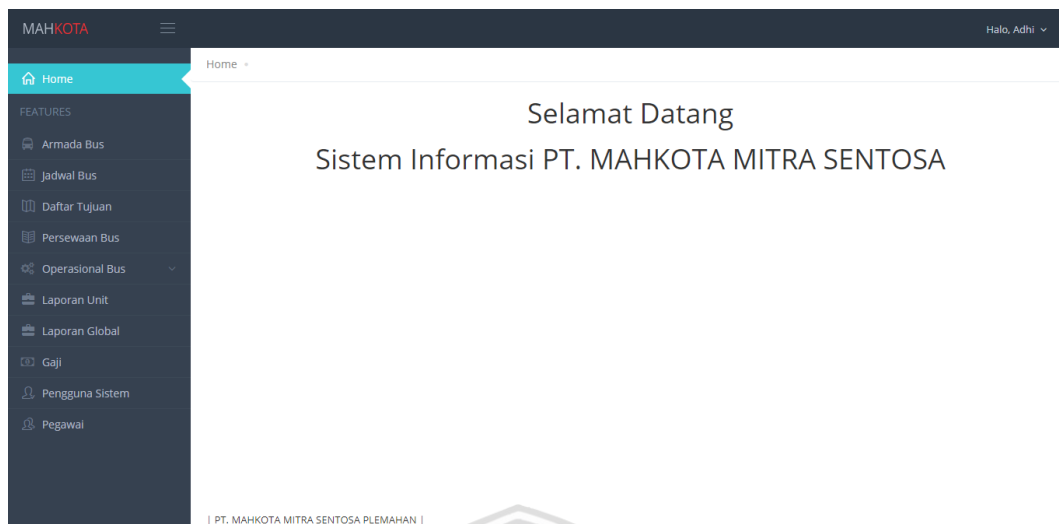
Antarmuka *login* ditampilkan kepada pengguna melalui peramban pada saat pengguna sistem menggunakan sistem dan harus melakukan autentikasi identitas pengguna. Antarmuka *login* yang terdapat dalam Gambar 6.1 menunjukkan beberapa komponen antarmuka di antaranya judul atau logo sistem informasi, kolom untuk memasukkan nama pengguna, kolom untuk memasukkan kata sandi pegawai, dan tombol untuk menjalankan fungsi autentikasi identitas pengguna sistem.



Gambar 6.1 Implementasi Antarmuka *Login*

### 6.3.2 Antarmuka *Home*

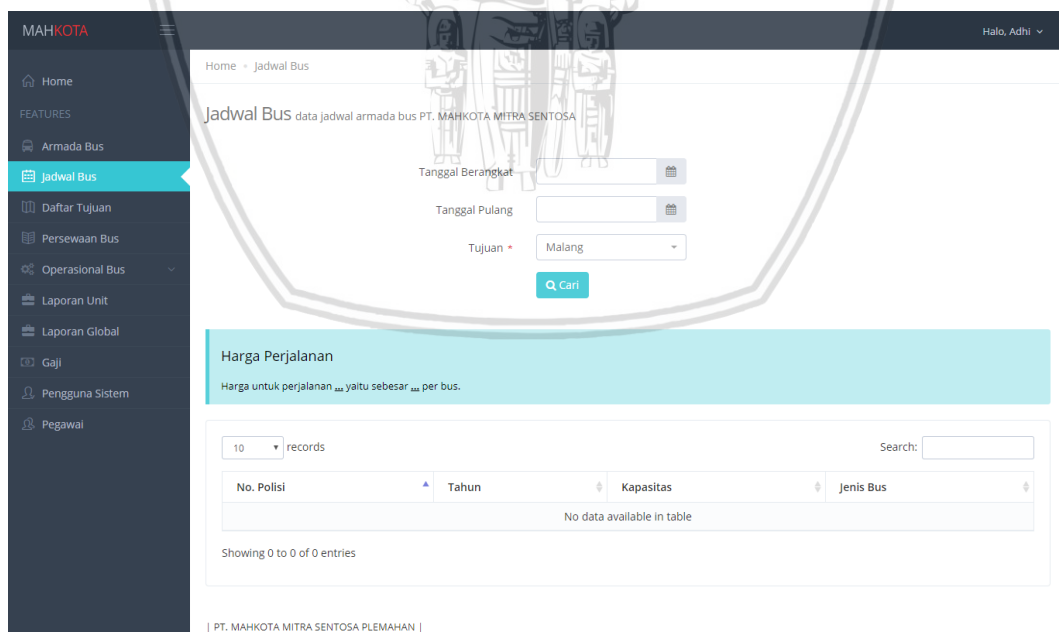
Antarmuka home ditampilkan kepada pengguna melalui peramban pada saat pengguna sistem telah melakukan autentikasi identitas pengguna. Antarmuka home yang terdapat dalam Gambar 6.2. Hasil implementasi antarmuka home menunjukkan komponen header yang berisi judul sistem informasi, informasi mengenai pengguna yang sedang menggunakan sistem, tombol *logout*, beberapa menu sistem pada sidebar, dan konten yang berisi tulisan “Selamat Datang Sistem Informasi PT MAHKOTA MITRA SENTOSA”.



**Gambar 6.2 Implementasi Antarmuka Home**

### 6.3.3 Antarmuka Jadwal Bus

Antarmuka jadwal bus ditampilkan kepada pengguna melalui peramban pada saat pengguna sistem menggunakan sistem untuk melihat jadwal bus. Antarmuka jadwal bus yang terdapat dalam Gambar 6.3. Hasil implementasi antarmuka jadwal bus menunjukkan komponen header yang berisi judul sistem informasi, informasi mengenai pengguna yang sedang menggunakan sistem, tombol *logout*, beberapa menu sistem pada sidebar, dan konten yang berisi *form* dan tabel.

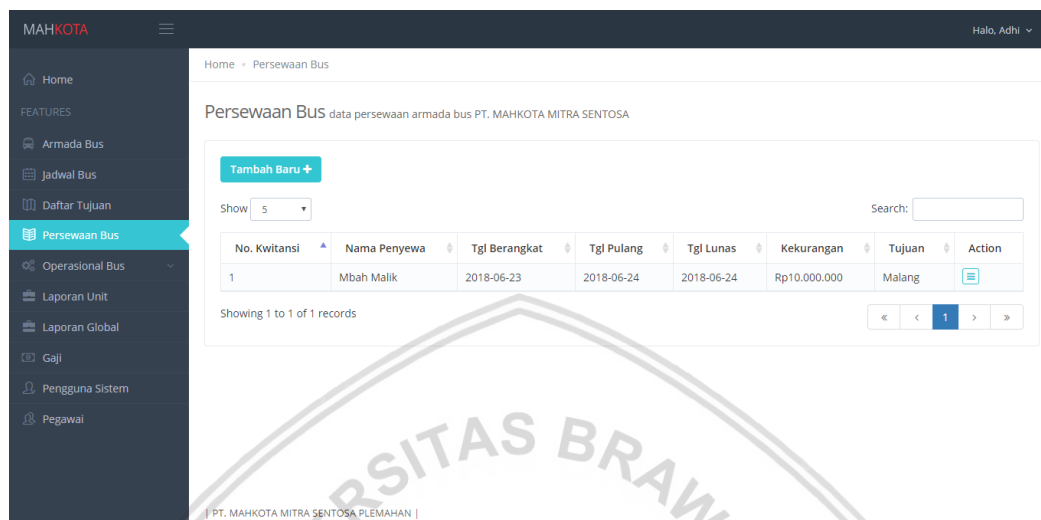


**Gambar 6.3 Implementasi Antarmuka Jadwal Bus**

### 6.3.4 Antarmuka Persewaan Bus

Antarmuka persewaan bus ditampilkan kepada pengguna melalui peramban pada saat pengguna sistem menggunakan sistem untuk melihat

persewaan bus. Antarmuka persewaan bus yang terdapat dalam Gambar 6.4. Hasil implementasi antarmuka persewaan bus menunjukkan komponen header yang berisi judul sistem informasi, informasi mengenai pengguna yang sedang menggunakan sistem, tombol *logout*, beberapa menu system pada sidebar, dan konten yang berisi tabel.



**Gambar 6.4 Implementasi Antarmuka Persewaan Bus**

### 6.3.5 Antarmuka Persewaan Bus: Menambah Persewaan

Antarmuka menambah persewaan bus ditampilkan kepada pengguna melalui peramban pada saat pengguna sistem menggunakan sistem untuk menambah persewaan bus. Antarmuka menambah persewaan bus yang terdapat dalam Gambar 6.5. Hasil implementasi antarmuka menambah persewaan bus menunjukkan komponen header yang berisi judul sistem informasi, informasi mengenai pengguna yang sedang menggunakan sistem, tombol *logout*, beberapa menu sistem pada sidebar, dan konten yang berisi *form*.



MAHKOTA

Halo, Adhi

Home » Persewaan Bus » Tambah Persewaan Bus

Tambah Persewaan Bus data armada bus PT. MAHKOTA MITRA SENTOSA

**TAMBAH PERSEWAAN BUS**

**Info Penyewa**

No Kwitansi \*  Nama Penyewa

No Handphone  Tempat Penjemputan

Jam Berangkat  Tujuan \*

Tgl Berangkat  Tgl Lunas

Tgl Pulang

**Info Armada**

No Polisi  Biaya Lain-lain

Harga Sewa  Sopir Cadangan

Telah Bayar  Kernet Cadangan

Fee  Catatan

Uang Makan

PT. MAHKOTA MITRA SENTOSA PLEMAHAN

Gambar 6.5 Implementasi Antarmuka Menambah Persewaan

### 6.3.6 Antarmuka Persewaan Bus: Menyunting Persewaan

Antarmuka menyunting persewaan bus ditampilkan kepada pengguna melalui peramban pada saat pengguna sistem menggunakan sistem untuk menyunting persewaan bus. Antarmuka menyunting persewaan bus yang terdapat dalam Gambar 6.6. Hasil implementasi antarmuka menyunting persewaan bus menunjukkan komponen header yang berisi judul sistem informasi, informasi mengenai pengguna yang sedang menggunakan sistem, tombol *logout*, beberapa menu sistem pada sidebar, dan konten yang berisi *form*.

MAHKOTA

Home • Persewaan Bus • Sunting Persewaan Bus

Sunting Persewaan Bus data persewaan bus PT. MAHKOTA MITRA SENTOSA

### SUNTING PERSEWAAN BUS

#### Info Penyewa

No Kwitansi • 14 Nama Penyewa sari

No Handphone • 08511122233 Tempat Penjemputan • kediri

Jam Berangkat 22:10 Tujuan • Malang

Tgl Berangkat 27-08-2018 Tgl Lunas 27-08-2018

Tgl Pulang 31-08-2018

#### Info Armada

No Polisi AG 7151 DU Biaya Lain-lain 0

Harga Sewa 1000000 Sopir Cadangan Tidak Pakai

Telah Bayar 0 Kernet Cadangan Tidak Pakai

Fee 0 Catatan (optional)

Uang Makan 200000

✖ Hapus Armada + Tambah Armada

Sunting Batal

PT. MAHKOTA MITRA SENTOSA PLEMAHAN

Gambar 6.6 Implementasi Antarmuka Menyunting Persewaan

### 6.3.7 Antarmuka Laporan Unit

Antarmuka laporan unit ditampilkan kepada pengguna melalui peramban pada saat pengguna sistem menggunakan sistem untuk melihat ataupun ingin melakukan cetak laporan unit. Antarmuka laporan unit yang terdapat dalam Gambar 6.7. Hasil implementasi antarmuka laporan unit menunjukkan komponen header yang berisi judul sistem informasi, informasi mengenai pengguna yang sedang menggunakan sistem, tombol *logout*, beberapa menu sistem pada sidebar, dan kontennya yang berisi tabel.

MAHKOTA

Home + Laporan Unit Bus

Laporan Unit Bus data persewaan armada bus PT. MAHKOTA MITRA SENTOSA

Periode 06/2018

No. Polisi AG 7151 DU

Cari

Print Excel PDF Columns

6 records

Search:

Penyewa	Tujuan	Jam	Tgl Berangkat	Tgl Pulang	Tgl Lunas	Penjemputan	Harga	Pot 5%
Mbah Malik	Malang	05:30:00	2018-06-23	2018-06-24	2018-09-21	Kediri	10000000	Rp500.000
Total							Rp10.000.000	Rp500.000

Showing 1 to 1 of 1 entries

PT. MAHKOTA MITRA SENTOSA PLEMAHAN

Gambar 6.7 Implementasi Antarmuka Laporan Unit

### 6.3.8 Antarmuka Laporan Global

Antarmuka laporan global ditampilkan kepada pengguna melalui peramban pada saat pengguna sistem menggunakan sistem untuk melihat ataupun ingin melakukan cetak laporan global. Antarmuka laporan global yang terdapat dalam Gambar 6.8. Hasil implementasi antarmuka laporan unit menunjukkan komponen header yang berisi judul sistem informasi, informasi mengenai pengguna yang sedang menggunakan sistem, tombol *logout*, beberapa menu sistem pada sidebar, dan konten yang berisi tabel.

MAHKOTA

Home + Laporan Global Bus

Laporan Global Bus data persewaan armada bus PT. MAHKOTA MITRA SENTOSA

Periode 06/2018

Cari

Print Excel Columns

10 records

Search:

No.	NOPOL	Nama	Harga	Fee+G.Bangku	Premi+Makan	Solar	Sisa
1	AG 7151 DU	Mbah Malik	Rp10.000.000	Rp0	Rp1.525.000	Rp2.000.000	5975000
Total			Rp10.000.000	Rp0	Rp1.525.000	Rp2.000.000	Rp5.975.000

Showing 1 to 1 of 1 entries

PT. MAHKOTA MITRA SENTOSA PLEMAHAN

Gambar 6.8 Implementasi Antarmuka Laporan Global

### 6.3.9 Antarmuka Gaji

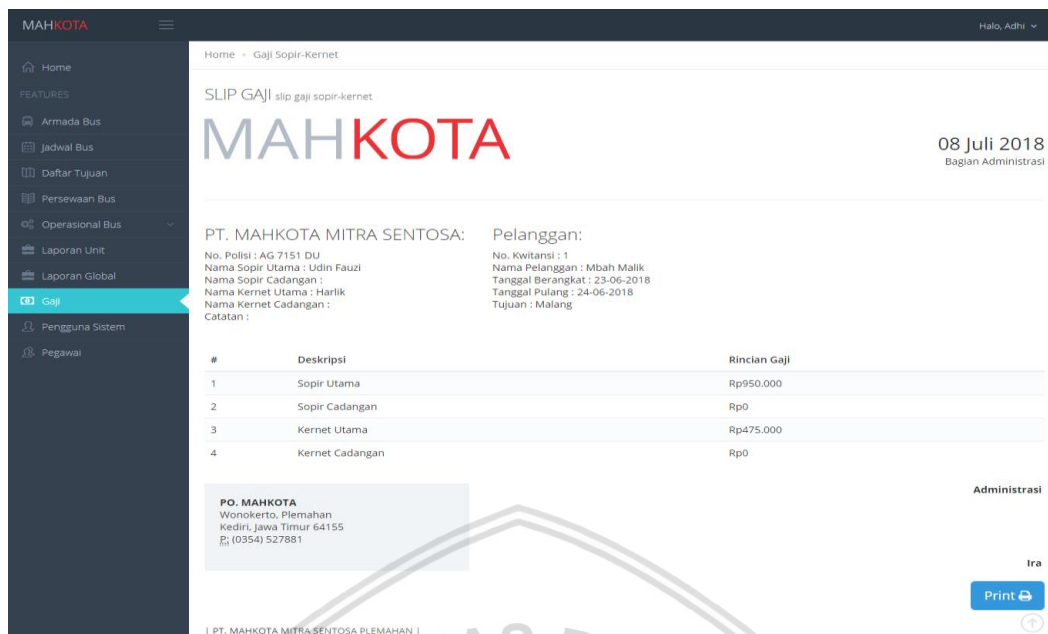
Antarmuka gaji ditampilkan kepada pengguna melalui peramban pada saat pengguna sistem menggunakan sistem untuk mencari data gaji. Antarmuka laporan unit yang terdapat dalam Gambar 6.7. Hasil implementasi

antarmuka gaji menunjukkan komponen header yang berisi judul sistem informasi, informasi mengenai pengguna yang sedang menggunakan sistem, tombol *logout*, beberapa menu sistem pada sidebar, dan konten yang berisi *form*.

**Gambar 6.9 Implementasi Antarmuka Gaji**

### 6.3.10 Antarmuka Gaji: Slip Gaji

Antarmuka slip gaji ditampilkan kepada pengguna melalui peramban pada saat pengguna sistem menggunakan sistem untuk melihat slip gaji ataupun ingin melakukan cetak slip gaji. Antarmuka slip gaji yang terdapat dalam Gambar 6.9. Hasil implementasi antarmuka slip gaji menunjukkan komponen header yang berisi judul sistem informasi, informasi mengenai pengguna yang sedang menggunakan sistem, tombol *logout*, beberapa menu sistem pada sidebar, dan konten yang berisi detail slip gaji.



**Gambar 6.10 Implementasi Antarmuka Slip Gaji**

## 6.4 Deployment

Subbab deployment menjelaskan fase transition pada RUP. Pada fase ini sistem informasi diimplementasikan melalui hosting. Adapun rincian *hosting* yang digunakan dijelaskan pada Tabel 6.1.

**Tabel 6.1 Rincian *Hosting***

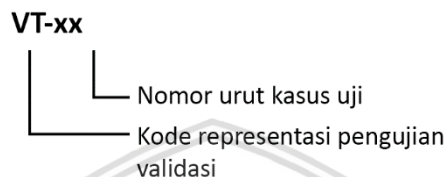
<b>Space</b>	2GB
<b>Bandwidth</b>	Unlimited
<b>Website</b>	Unlimited
<b>CPU (Core)</b>	1
<b>RAM (MB)</b>	768
<b>Multiple PHP (4, 5.2, 5.3, 5.4, 5.5, 5.6, 7)</b>	Iya
<b>MySQL</b>	Iya

Pada Tabel 6.1 dijelaskan *hosting* yang dipakai menggunakan hosting yang mempunyai kapasitas *space* 2GB, *bandwidth* unlimited, *website* unlimited, *multiple PHP* (4, 5.2, 5.3, 5.4, 5.5, 5.6, 7) tersedia, dan *MySQL* tersedia. Kapasitas yang ada pada *hosting* telah mendukung untuk mengimplementasikan sistem informasi. Semua file hasil pengembangan yang ada pada *localhost* diunggah pada hosting di folder publik\_html. Kemudian untuk *database* dilakukan *import* melalui admin panel yang tersedia. Sistem informasi yang telah diimplementasikan dapat dibuka melalui alamat URL <https://tirenn.site/mahkotabus/>.

## BAB 7 PENGUJIAN

### 7.1 Pengujian *Black-Box*

Pengujian *black-box* dilakukan untuk mengetahui bahwa sistem dapat berjalan sesuai dengan ekspektasi pengguna. Subbab ini berisi hasil pengujian *black-box* terhadap beberapa fungsi sistem informasi yang digunakan oleh pengguna sesuai dengan kebutuhan fungsional yang telah didefinisikan pada bab sebelumnya.



**Gambar 7.1 Kodifikasi Pengujian *Black-Box***

Pengujian *black-box* dilakukan berdasarkan skenario uji yang diperoleh dari alur *use-case* yang telah didefinisikan pada bab sebelumnya, kasus uji dapat diidentifikasi berdasarkan skenario uji yang telah ditentukan dan pengujian *black-box* dapat dilakukan. Kasus uji *black-box* memiliki kodifikasi yang ditunjukkan pada Gambar 7.1.

#### 7.1.1 Pengujian *Black-Box* Melakukan *Login*

Bagian ini menjelaskan pengujian *black-box* pada beberapa fungsi yang dapat digunakan oleh pengguna untuk melakukan *login* pada sistem yang telah dikembangkan. Pada Tabel 7.1 menunjukkan beberapa skenario yang bersumber dari alur *use-case* Melakukan *Login* yang akan digunakan untuk mengidentifikasi kasus uji.

**Tabel 7.1 Pengujian *Black-Box* Melakukan *Login***

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	<i>Basic flow</i>	
Skenario 2	<i>Basic flow</i>	A1. Unidentified User

##### 7.1.1.1 Kasus Uji Validasi Melakukan *Login*

Pada Tabel 7.2 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-01. Kasus uji VT-01 dilakukan untuk memastikan bahwa sistem dapat menjalankan fungsi *login* dengan kode persyaratan SKPL-F-01. Kasus uji diidentifikasi berdasarkan skenario 1 pada Tabel 7.1. Pengujian *black-box* Melakukan *Login* menunjukkan hasil valid.

**Tabel 7.2 Kasus Uji Validasi Melakukan *Login***



<b>Kode Pengujian</b>	VT-01
<b>Nomor Skenario</b>	Skenario 1
<b>Kode Persyaratan</b>	SKPL-F-01
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menjalankan fungsi <i>login</i>
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji teridentifikasi memiliki akun pengguna sistem.</li> <li>2. Penguji mengisi <i>form login</i> sesuai dengan data akun.</li> <li>3. Penguji memilih tombol <i>login</i>.</li> </ol>
<b>Hasil yang Diharapkan</b>	Sistem menampilkan halaman utama sistem informasi tanpa ada pesan kesalahan.
<b>Hasil Pengujian</b>	Sistem menampilkan halaman utama sistem informasi tanpa ada pesan kesalahan.
<b>Status Uji</b>	valid

#### 7.1.1.2 Kasus Uji Validasi Melakukan Login: A1 Unidentified User

Pada Tabel 7.3 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-02. Kasus uji VT-01 dilakukan untuk memastikan bahwa sistem dapat menjalankan fungsi *login* dengan kode persyaratan SKPL-F-01. Kasus uji diidentifikasi berdasarkan skenario 1 pada Tabel 7.1. Pengujian *black-box* Melakukan Login menunjukkan hasil valid.

**Tabel 7.3 Kasus Uji Validasi Melakukan Login: A1 Unidentified User**

<b>Kode Pengujian</b>	VT-02
<b>Nomor Skenario</b>	Skenario 2
<b>Kode Persyaratan</b>	SKPL-F-01
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menjalankan fungsi <i>login</i>
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji teridentifikasi tidak memiliki akun pengguna sistem.</li> <li>2. Penguji mengisi <i>form login</i>.</li> <li>3. Penguji memilih tombol <i>login</i>.</li> </ol>
<b>Hasil yang Diharapkan</b>	Sistem menampilkan pesan bahwa penguji tidak dapat masuk ke sistem karena tidak ada akun yang sesuai dengan masukkan penguji.

<b>Hasil Pengujian</b>	Sistem menampilkan pesan bahwa penguji tidak dapat masuk ke sistem karena tidak ada akun yang sesuai dengan masukkan penguji.
<b>Status Uji</b>	Valid

### 7.1.2 Pengujian *Black-Box* Melakukan *Logout*

Bagian ini menjelaskan pengujian *black-box* pada beberapa fungsi yang dapat digunakan oleh pengguna untuk melakukan *logout* pada sistem yang telah dikembangkan. Pada Tabel 7.4 menunjukkan beberapa skenario yang bersumber dari alur *use-case* Melakukan *Logout* yang akan digunakan untuk mengidentifikasi kasus uji.

**Tabel 7.4 Pengujian *Black-Box* Melakukan *Logout***

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	<i>Basic flow</i>	

#### 7.1.2.1 Kasus Uji Validasi Melakukan *Logout*

Pada Tabel 7.5 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-03. Kasus uji VT-03 dilakukan untuk memastikan bahwa sistem dapat menjalankan fungsi *logout* dengan kode persyaratan SKPL-F-02. Kasus uji diidentifikasi berdasarkan skenario 1 pada Tabel 7.4. Pengujian *black-box* Melakukan *Logout* menunjukkan hasil valid.

**Tabel 7.5 Kasus Uji Validasi Melakukan *Logout***

<b>Kode Pengujian</b>	VT-03
<b>Nomor Skenario</b>	Skenario 1
<b>Kode Persyaratan</b>	SKPL-F-02
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menjalankan fungsi <i>logout</i>
<b>Prosedur Uji</b>	1. Penguji telah masuk ke dalam sistem. 2. Penguji memilih tombol <i>logout</i> .
<b>Hasil yang Diharapkan</b>	Sistem menampilkan halaman <i>login</i> .
<b>Hasil Pengujian</b>	Sistem menampilkan halaman <i>login</i> .
<b>Status Uji</b>	Valid

### 7.1.3 Pengujian *Black-Box* Melihat Jadwal Armada Bus

Bagian ini menjelaskan pengujian *black-box* pada beberapa fungsi yang dapat digunakan oleh pengguna untuk melihat jadwal armada bus melalui sistem yang telah dikembangkan. Pada Tabel 7.6 menunjukkan beberapa skenario yang bersumber dari alur usecase Melihat Jadwal Armada Bus yang akan digunakan untuk mengidentifikasi kasus uji.

**Tabel 7.6 Pengujian *Black-Box* Melihat Jadwal Armada Bus**

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	<i>Basic flow</i>	

#### 7.1.3.1 Kasus Uji Validasi Melihat Jadwal Armada Bus

Pada Tabel 7.7 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-04. Kasus uji VT-04 dilakukan untuk memastikan bahwa sistem dapat menjalankan fungsi *logout* dengan kode persyaratan SKPL-F-03. Kasus uji diidentifikasi berdasarkan skenario 1 pada Tabel 7.6. Pengujian *black-box* Melakukan *Logout* menunjukkan hasil valid.

**Tabel 7.7 Kasus Uji Validasi Melihat Jadwal Armada Bus**

Kode Pengujian	VT-04
Nomor Skenario	Skenario 1
Kode Persyaratan	SKPL-F-03
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat menampilkan jadwal bus yang tersedia.
Prosedur Uji	<ol style="list-style-type: none"> <li>1. Penguji telah masuk ke dalam sistem.</li> <li>2. Penguji memilih menu jadwal bus.</li> <li>3. Penguji memasukkan data pada <i>form</i> yang telah tersedia.</li> <li>4. Penguji memilih tombol cari.</li> </ol>
Hasil yang Diharapkan	Sistem menampilkan data jadwal bus yang tersedia.
Hasil Pengujian	Sistem menampilkan data jadwal bus yang tersedia.
Status Uji	Valid

### 7.1.4 Pengujian *Black-Box* Mengelola Armada Bus

Bagian ini menjelaskan pengujian *black-box* pada beberapa fungsi yang dapat digunakan oleh pengguna untuk mengelola armada bus melalui sistem

yang telah dikembangkan. Pada Tabel 7.8 menunjukkan beberapa skenario yang bersumber dari alur *use-case* Mengelola Armada Bus yang akan digunakan untuk mengidentifikasi kasus uji.

**Tabel 7.8 Pengujian *Black-Box* Mengelola Armada Bus**

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	<i>Basic flow</i>	
Skenario 2	<i>Basic flow</i>	A1: Menghapus armada bus
Skenario 3	<i>Basic flow</i>	A2: Menyunting armada bus
Skenario 4	<i>Basic flow</i>	A3: <i>Incomplete Form</i>

#### 7.1.4.1 Kasus Uji Validasi Mengelola Armada Bus

Pada Tabel 7.9 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-05. Kasus uji VT-05 dilakukan untuk memastikan bahwa sistem dapat menambah data armada bus baru dengan kode persyaratan SKPL-F-04. Kasus uji diidentifikasi berdasarkan skenario 1 pada Tabel 7.8. Pengujian *black-box* Mengelola Armada bus menunjukkan hasil valid.

**Tabel 7.9 Kasus Uji Validasi Mengelola Armada Bus**

Kode Pengujian	VT-05
Nomor Skenario	Skenario 1
Kode Persyaratan	SKPL-F-04
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat menampilkan armada bus yang tersedia dan dapat menambah data armada bus baru.
Prosedur Uji	<ol style="list-style-type: none"> <li>1. Penguji telah masuk ke dalam sistem.</li> <li>2. Penguji memilih menu armada bus.</li> <li>3. Penguji memilih menu tambah.</li> <li>4. Penguji mengisi <i>form</i> yang telah disediakan dengan benar.</li> </ol>
Hasil yang Diharapkan	Sistem menampilkan halaman armada bus dengan data yang tersedia dan sistem dapat menyimpan data armada bus baru yang telah ditambahkan.
Hasil Pengujian	Sistem menampilkan halaman armada bus dengan data yang tersedia dan sistem dapat menyimpan data armada bus baru yang telah ditambahkan.

Status Uji	Valid
------------	-------

#### 7.1.4.2 Kasus Uji Validasi Mengelola Armada Bus: A1 Menghapus Armada Bus

Pada Tabel 7.10 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-06. Kasus uji VT-06 dilakukan untuk memastikan bahwa sistem dapat menghapus armada bus dengan kode persyaratan SKPL-F-05. Kasus uji diidentifikasi berdasarkan skenario 2 pada Tabel 7.8. Pengujian *black-box* Menghapus Armada bus menunjukkan hasil valid.

**Tabel 7.10 Kasus Uji Validasi Mengelola Armada Bus: A1 Menghapus Armada Bus**

Kode Pengujian	VT-06
Nomor Skenario	Skenario 2
Kode Persyaratan	SKPL-F-05
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat menghapus armada bus yang tersedia.
Prosedur Uji	<ol style="list-style-type: none"> <li>1. Penguji telah masuk ke dalam sistem.</li> <li>2. Penguji memilih menu armada bus.</li> <li>3. Penguji memilih tombol <i>delete</i>.</li> <li>4. Penguji memilih tombol Yes pada pemberitahuan konfirmasi.</li> </ol>
Hasil yang Diharapkan	Sistem menampilkan data armada bus yang tidak dihapus.
Hasil Pengujian	Sistem menampilkan data armada bus yang tidak dihapus.
Status Uji	Valid

#### 7.1.4.3 Kasus Uji Validasi Mengelola Armada Bus: A2 Menyunting Armada Bus

Pada Tabel 7.11 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-07. Kasus uji VT-07 dilakukan untuk memastikan bahwa sistem dapat menyunting armada bus dengan kode persyaratan SKPL-F-06. Kasus uji diidentifikasi berdasarkan skenario 3 pada Tabel 7.8. Pengujian *black-box* Menyunting Armada bus menunjukkan hasil valid.

**Tabel 7.11 Kasus Uji Validasi Mengelola Armada Bus: A2 Menyunting Armada Bus**

Kode Pengujian	VT-07
----------------	-------

<b>Nomor Skenario</b>	Skenario 3
<b>Kode Persyaratan</b>	SKPL-F-06
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menyunting armada bus yang tersedia.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah masuk ke dalam sistem.</li> <li>2. Penguji memilih menu armada bus.</li> <li>3. Penguji memilih tombol <i>edit</i>.</li> <li>4. Penguji memasukkan data pada <i>form</i> yang telah ditampilkan oleh sistem dengan benar.</li> </ol>
<b>Hasil yang Diharapkan</b>	Sistem menampilkan pesan bahwa data telah disimpan.
<b>Hasil Pengujian</b>	Sistem menampilkan pesan bahwa data telah disimpan.
<b>Status Uji</b>	Valid

#### 7.1.4.4 Kasus Uji Validasi Mengelola Armada Bus: A3 *Incomplete Form*

Pada Tabel 7.12 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-08. Kasus uji VT-08 dilakukan untuk memastikan bahwa sistem dapat menampilkan pesan bahwa *form* belum tersisi secara penuh dan sesuai. Kasus uji diidentifikasi berdasarkan skenario 1 pada Tabel 7.8. Pengujian *black-box Incomplete Form* pada Mengelola Armada bus menunjukkan hasil valid.

**Tabel 7.12 Kasus Uji Validasi Mengelola Armada Bus: A3 *Incomplete Form***

<b>Kode Pengujian</b>	VT-08
<b>Nomor Skenario</b>	Skenario 2
<b>Kode Persyaratan</b>	-
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat memberikan pesan bahwa <i>form</i> pengisian belum terisi secara penuh dan sesuai.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah masuk ke dalam sistem.</li> <li>2. Penguji telah masuk ke dalam halaman tambah armada atau sunting armada.</li> <li>3. Penguji tidak mengisi <i>form</i> secara penuh dan sesuai.</li> <li>4. Penguji memilih tombol tambah atau sunting (sesuai halaman).</li> </ol>
<b>Hasil yang Diharapkan</b>	Sistem menampilkan pesan bahwa <i>form</i> masih ada salah.



<b>Hasil Pengujian</b>	Sistem menampilkan pesan bahwa <i>form</i> masih ada salah.
<b>Status Uji</b>	Valid

### 7.1.5 Pengujian *Black-Box* Mengelola Persewaan Bus

Bagian ini menjelaskan pengujian *black-box* pada beberapa fungsi yang dapat digunakan oleh pengguna untuk mengelola persewaan bus melalui sistem yang telah dikembangkan. Pada Tabel 7.13 menunjukkan beberapa skenario yang bersumber dari alur *use-case* Mengelola Persewaan Bus yang akan digunakan untuk mengidentifikasi kasus uji.

**Tabel 7.13 Pengujian *Black-Box* Mengelola Persewaan Bus**

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	<i>Basic flow</i>	
Skenario 2	<i>Basic flow</i>	A1. Menyunting Persewaan Bus

#### 7.1.5.1 Kasus Uji Validasi Mengelola Persewaan

Pada Tabel 7.14 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-09. Kasus uji VT-09 dilakukan untuk memastikan bahwa sistem dapat menambahkan data persewaan baru dengan kode persyaratan SKPL-F-10. Kasus uji diidentifikasi berdasarkan skenario 1 pada Tabel 7.13. Pengujian *black-box* Mengelola Persewaan menunjukkan hasil valid.

**Tabel 7.14 Kasus Uji Validasi Mengelola Persewaan**

<b>Kode Pengujian</b>	VT-09
<b>Nomor Skenario</b>	Skenario 1
<b>Kode Persyaratan</b>	SKPL-F-10
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menampilkan data persewaan yang telah tercatat dan sistem dapat menyimpan data persewaan baru.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah masuk ke dalam sistem.</li> <li>2. Penguji memilih menu persewaan bus.</li> <li>3. Penguji memilih menu tambah.</li> <li>4. Penguji mengisi <i>form</i> yang telah disediakan dengan benar.</li> </ol>
<b>Hasil yang Diharapkan</b>	System menampilkan halaman persewaan dengan data persewaan yang tersedia dan sistem dapat menyimpan data persewaan baru yang telah dimasukkan.

<b>Hasil Pengujian</b>	System menampilkan halaman persewaan dengan data persewaan yang tersedia dan sistem dapat menyimpan data persewaan baru yang telah dimasukkan.
<b>Status Uji</b>	Valid

#### 7.1.5.2 Kasus Uji Validasi Mengelola Persewaan: A1 Menyunting Persewaan Bus

Pada Tabel 7.15 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-10. Kasus uji VT-10 dilakukan untuk memastikan bahwa sistem dapat menyunting data persewaan yang telah tercatat dengan kode persyaratan SKPL-F-11. Kasus uji diidentifikasi berdasarkan skenario 2 pada Tabel 7.13. Pengujian *black-box* Menyunting Persewaan Bus menunjukkan hasil valid.

**Tabel 7.15 Kasus Uji Validasi Mengelola Persewaan: A1 Menyunting Persewaan Bus**

<b>Kode Pengujian</b>	VT-10
<b>Nomor Skenario</b>	Skenario 2
<b>Kode Persyaratan</b>	SKPL-F-11
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menyunting data persewaan yang telah tercatat.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah masuk ke dalam sistem.</li> <li>2. Penguji memilih menu persewaan.</li> <li>3. Penguji memilih tombol edit.</li> <li>4. Penguji mengisi <i>form</i> yang telah disediakan.</li> <li>5. Penguji memilih tombol sunting.</li> </ol>
<b>Hasil yang Diharapkan</b>	System menampilkan halaman sunting persewaan. data persewaan dan sistem dapat menyimpan perubahan data.
<b>Hasil Pengujian</b>	System menampilkan halaman sunting persewaan. data persewaan dan sistem dapat menyimpan perubahan data.
<b>Status Uji</b>	Valid

#### 7.1.5.3 Kasus Uji Validasi Mengelola Persewaan: A2 *Incomplete Form*

Pada Tabel 7.16 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-11. Kasus uji VT-11 dilakukan untuk memastikan bahwa sistem dapat menampilkan pesan bahwa *form* belum tersisi secara penuh dan sesuai. Kasus uji diidentifikasi berdasarkan skenario 1 pada Tabel 7.13.

Pengujian *black-box Incomplete Form* pada Mengelola Persewaan Bus menunjukkan hasil valid.

**Tabel 7.16 Kasus Uji Validasi Mengelola Persewaan: A2 *Incomplete Form***

<b>Kode Pengujian</b>	VT-11
<b>Nomor Skenario</b>	Skenario 2
<b>Kode Persyaratan</b>	-
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat memberikan pesan bahwa <i>form</i> pengisian belum terisi secara penuh dan sesuai.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah masuk ke dalam sistem.</li> <li>2. Penguji telah masuk ke dalam halaman tambah persewaan atau sunting persewaan.</li> <li>3. Penguji tidak mengisi <i>form</i> secara penuh dan sesuai.</li> <li>4. Penguji memilih tombol tambah atau sunting (sesuai halaman).</li> </ol>
<b>Hasil yang Diharapkan</b>	Sistem menampilkan pesan bahwa <i>form</i> masih ada salah.
<b>Hasil Pengujian</b>	Sistem menampilkan pesan bahwa <i>form</i> masih ada salah.
<b>Status Uji</b>	Valid

### 7.1.6 Pengujian *Black-Box* Mencetak Laporan Unit

Bagian ini menjelaskan pengujian *black-box* pada beberapa fungsi yang dapat digunakan oleh pengguna untuk melihat dan mencetak laporan unit melalui sistem yang telah dikembangkan. Pada Tabel 7.17 menunjukkan beberapa skenario yang bersumber dari alur *use-case* Melihat Laporan Unit dan Mencetak Laporan Unit yang akan digunakan untuk mengidentifikasi kasus uji.

**Tabel 7.17 Pengujian *Black-Box* Mencetak Laporan Unit**

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	<i>Basic flow</i>	

#### 7.1.6.1 Kasus Uji Validasi Mencetak Laporan Unit

Pada Tabel 7.18 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-12. Kasus uji VT-12 dilakukan untuk memastikan bahwa sistem dapat menampilkan data laporan unit persewaan yang telah tercatat dengan kode persyaratan SKPL-F-13. Kasus uji diidentifikasi berdasarkan skenario

1 pada Tabel 7.17. Pengujian *black-box* Mencetak Laporan Unit menunjukkan hasil valid.

**Tabel 7.18 Kasus Uji Validasi Mencetak Laporan Unit**

<b>Kode Pengujian</b>	VT-12
<b>Nomor Skenario</b>	Skenario 1
<b>Kode Persyaratan</b>	SKPL-F-13
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menampilkan data laporan unit persewaan yang telah tercatat.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah masuk ke dalam sistem.</li> <li>2. Penguji memilih menu laporan unit.</li> <li>3. Penguji mengisi <i>form</i> dengan benar.</li> <li>4. Penguji memilih tombol print untuk mencetak.</li> </ol>
<b>Hasil yang Diharapkan</b>	System menampilkan data laporan unit pada periode, unit armada sesuai dengan masukkan pada <i>form</i> , dan sistem dapat menampilkan halaman cetak.
<b>Hasil Pengujian</b>	System menampilkan data laporan unit pada periode, unit armada sesuai dengan masukkan pada <i>form</i> , dan sistem dapat menampilkan halaman cetak.
<b>Status Uji</b>	Valid

### 7.1.7 Pengujian *Black-Box* Mencetak Laporan Global

Bagian ini menjelaskan pengujian *black-box* pada beberapa fungsi yang dapat digunakan oleh pengguna untuk melihat dan mencetak laporan global melalui sistem yang telah dikembangkan. Pada Tabel 7.19 menunjukkan beberapa skenario yang bersumber dari alur *use-case* Melihat Laporan Global dan Mencetak Laporan Global yang akan digunakan untuk mengidentifikasi kasus uji.

**Tabel 7.19 Pengujian *Black-Box* Mencetak Laporan Global**

<b>No. Skenario</b>	<b>Alur Awal</b>	<b>Alur Alternatif</b>
Skenario 1	<i>Basic flow</i>	

#### 7.1.7.1 Kasus Uji Validasi Mencetak Laporan Global

Pada Tabel 7.20 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-13. Kasus uji VT-13 dilakukan untuk memastikan bahwa sistem dapat menampilkan data laporan global persewaan yang telah tercatat dengan kode persyaratan SKPL-F-14. Kasus uji diidentifikasi berdasarkan

skenario 1 pada Tabel 7.19. Pengujian *black-box* Mencetak Laporan Global menunjukkan hasil valid.

**Tabel 7.20 Kasus Uji Validasi Mencetak Laporan Global**

<b>Kode Pengujian</b>	VT-13
<b>Nomor Skenario</b>	Skenario 1
<b>Kode Persyaratan</b>	SKPL-F-14
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menampilkan data laporan unit persewaan yang telah tercatat.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah masuk ke dalam sistem.</li> <li>2. Penguji memilih menu laporan global.</li> <li>3. Penguji mengisi <i>form</i> dengan benar.</li> <li>4. Penguji memilih tombol print untuk mencetak.</li> </ol>
<b>Hasil yang Diharapkan</b>	System menampilkan data laporan global pada periode sesuai dengan masukkan pada <i>form</i> , dan sistem dapat menampilkan halaman cetak.
<b>Hasil Pengujian</b>	System menampilkan data laporan global pada periode sesuai dengan masukkan pada <i>form</i> , dan sistem dapat menampilkan halaman cetak.
<b>Status Uji</b>	Valid

### 7.1.8 Pengujian *Black-Box* Mencetak Slip Gaji

Bagian ini menjelaskan pengujian *black-box* pada beberapa fungsi yang dapat digunakan oleh pengguna untuk melihat dan mencetak slip gaji melalui sistem yang telah dikembangkan. Pada Tabel 7.21 menunjukkan beberapa skenario yang bersumber dari alur *use-case* Melihat Slip Gaji dan Mencetak Slip Gaji yang akan digunakan untuk mengidentifikasi kasus uji.

**Tabel 7.21 Pengujian *Black-Box* Mencetak Slip Gaji**

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	<i>Basic flow</i>	
Skenario 2	<i>Basic flow</i>	A1: Data tidak ditemukan

#### 7.1.8.1 Kasus Uji Validasi Mencetak Slip Gaji

Pada Tabel 7.22 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-14. Kasus uji VT-14 dilakukan untuk memastikan bahwa sistem dapat mencetak slip gaji dengan kode persyaratan SKPL-F-15. Kasus uji diidentifikasi berdasarkan skenario 1 pada Tabel 7.21. Pengujian *black-box* Melihat Slip Gaji dan Mencetak Slip Gaji menunjukkan hasil valid.

**Tabel 7.22 Kasus Uji Validasi Mencetak Slip Gaji**

<b>Kode Pengujian</b>	VT-14
<b>Nomor Skenario</b>	Skenario 1
<b>Kode Persyaratan</b>	SKPL-F-15
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menampilkan data slip gaji persewaan yang telah tercatat dan dapat mencetak slip gaji.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah masuk ke dalam sistem.</li> <li>2. Penguji memilih menu gaji.</li> <li>3. Penguji mengisi <i>form</i> dengan benar.</li> <li>4. Penguji memilih tombol cari.</li> <li>5. Penguji memilih tombol cetak untuk mencetak.</li> </ol>
<b>Hasil yang Diharapkan</b>	System menampilkan data slip gaji yang sesuai dengan masukkan pada <i>form</i> , dan sistem dapat menampilkan halaman cetak.
<b>Hasil Pengujian</b>	System menampilkan data slip gaji yang sesuai dengan masukkan pada <i>form</i> , dan sistem dapat menampilkan halaman cetak.
<b>Status Uji</b>	Valid

#### 7.1.8.2 Kasus Uji Validasi Mencetak Slip Gaji: A1 Data tidak ditemukan

Pada Tabel 7.23 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-15. Kasus uji VT-15 dilakukan untuk memastikan bahwa sistem dapat menampilkan pesan jika data tidak ditemukan pada saat pencarian data. Kasus uji diidentifikasi berdasarkan skenario 1 pada Tabel 7.21. Pengujian *black-box* alternative Data Tidak Ditemukan pada Mencetak Slip Gaji menunjukkan hasil valid.

**Tabel 7.23 Kasus Uji Validasi Mencetak Slip Gaji: A1 Data tidak ditemukan**

<b>Kode Pengujian</b>	VT-15
<b>Nomor Skenario</b>	Skenario 2



<b>Kode Persyaratan</b>	-
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menampilkan pesan data tidak ditemukan pada saat pencarian data.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah masuk ke dalam sistem.</li> <li>2. Penguji memilih menu gaji.</li> <li>3. Penguji mengisi <i>form</i> dengan dengan no kuitansi salah.</li> <li>4. Penguji memilih tombol cari.</li> </ol>
<b>Hasil yang Diharapkan</b>	System menampilkan pesan data tidak ditemukan setelah tombol cari dipilih.
<b>Hasil Pengujian</b>	System menampilkan pesan data tidak ditemukan setelah tombol cari dipilih.
<b>Status Uji</b>	Valid

### 7.1.9 Pengujian *Black-Box* Mengelola Pegawai

Bagian ini menjelaskan pengujian *black-box* pada beberapa fungsi yang dapat digunakan oleh pengguna untuk mengelola pegawai melalui sistem yang telah dikembangkan. Pada Tabel 7.24 menunjukkan beberapa skenario yang bersumber dari alur *use-case* Mengelola Pegawai yang akan digunakan untuk mengidentifikasi kasus uji.

**Tabel 7.24 Pengujian *Black-Box* Mengelola Pegawai**

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	<i>Basic flow</i>	
Skenario 2	<i>Basic flow</i>	A1. Menghapus Persewaan Bus
Skenario 3	<i>Basic flow</i>	A2. Menyunting Persewaan Bus
Skenario 4	<i>Basic flow</i>	A3 <i>Incomplete Form</i>

#### 7.1.9.1 Kasus Uji Validasi Mengelola Pegawai

Pada Tabel 7.25 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-16. Kasus uji VT-16 dilakukan untuk memastikan bahwa sistem dapat menambahkan data pegawai baru dengan kode persyaratan SKPL-F-16. Kasus uji diidentifikasi berdasarkan skenario 1 pada Tabel 7.24. Pengujian *black-box* Mengelola Pegawai menunjukkan hasil valid.

Tabel 7.25 Kasus Uji Validasi Mengelola Pegawai

<b>Kode Pengujian</b>	VT-16
<b>Nomor Skenario</b>	Skenario 1
<b>Kode Persyaratan</b>	SKPL-F-16
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menampilkan data pegawai yang telah tercatat dan sistem dapat menyimpan data pegawai baru.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah masuk ke dalam sistem.</li> <li>2. Penguji memilih menu pegawai.</li> <li>3. Penguji memilih menu tambah.</li> <li>4. Penguji mengisi <i>form</i> yang telah disediakan dengan benar.</li> </ol>
<b>Hasil yang Diharapkan</b>	System menampilkan halaman pegawai dengan data pegawai yang tersedia dan sistem dapat menyimpan data pegawai baru yang telah dimasukkan.
<b>Hasil Pengujian</b>	System menampilkan halaman pegawai dengan data pegawai yang tersedia dan sistem dapat menyimpan data pegawai baru yang telah dimasukkan.
<b>Status Uji</b>	Valid

#### 7.1.9.2 Kasus Uji Validasi Mengelola Pegawai: A1 Menghapus Pegawai

Pada Tabel 7.26 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-17. Kasus uji VT-17 dilakukan untuk memastikan bahwa sistem dapat menghapus pegawai dengan kode persyaratan SKPL-F-17. Kasus uji diidentifikasi berdasarkan skenario 2 pada Tabel 7.24. Pengujian *black-box* Menghapus Pengguna Sistem menunjukkan hasil valid.

Tabel 7.26 Kasus Uji Validasi Mengelola Pegawai: A1 Menghapus Pegawai

<b>Kode Pengujian</b>	VT-17
<b>Nomor Skenario</b>	Skenario 2
<b>Kode Persyaratan</b>	SKPL-F-17
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menghapus pegawai yang tersedia.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah masuk ke dalam sistem.</li> <li>2. Penguji memilih menu pegawai.</li> <li>3. Penguji memilih tombol <i>delete</i>.</li> </ol>

	4. Penguji memilih tombol <i>Yes</i> pada pemberitahuan konfirmasi.
<b>Hasil yang Diharapkan</b>	Sistem menampilkan data pegawai yang tidak dihapus.
<b>Hasil Pengujian</b>	Sistem menampilkan data pegawai yang tidak dihapus.
<b>Status Uji</b>	Valid

#### 7.1.9.3 Kasus Uji Validasi Mengelola Pegawai: A2 Menyunting Pegawai

Pada Tabel 7.27 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-18. Kasus uji VT-18 dilakukan untuk memastikan bahwa sistem dapat menyunting data pegawai yang telah tercatat dengan kode persyaratan SKPL-F-18. Kasus uji diidentifikasi berdasarkan skenario 3 pada Tabel 7.24. Pengujian *black-box* Menyunting Pegawai menunjukkan hasil valid.

**Tabel 7.27 Kasus Uji Validasi Mengelola Pegawai: A2 Menyunting Pegawai**

<b>Kode Pengujian</b>	VT-18
<b>Nomor Skenario</b>	Skenario 3
<b>Kode Persyaratan</b>	SKPL-F-18
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menyunting data pegawai yang telah tercatat.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah masuk ke dalam sistem.</li> <li>2. Penguji memilih menu persewaan.</li> <li>3. Penguji memilih tombol edit.</li> <li>4. Penguji mengisi <i>form</i> yang telah disediakan.</li> <li>5. Penguji memilih tombol sunting.</li> </ol>
<b>Hasil yang Diharapkan</b>	System menampilkan halaman sunting pegawai, data pegawai dan sistem dapat menyimpan perubahan.
<b>Hasil Pengujian</b>	System menampilkan halaman sunting pegawai, data pegawai dan sistem dapat menyimpan perubahan.
<b>Status Uji</b>	Valid

#### 7.1.9.4 Kasus Uji Validasi Mengelola Pegawai: A3 *Incomplete Form*

Pada Tabel 7.28 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-19. Kasus uji VT-19 dilakukan untuk memastikan bahwa sistem dapat menampilkan pesan bahwa *form* belum tersisi secara penuh dan sesuai. Kasus uji diidentifikasi berdasarkan skenario 4 pada Tabel 7.24.

Pengujian *black-box Incomplete Form* pada Mengelola Pegawai menunjukkan hasil valid.

**Tabel 7.28 Kasus Uji Validasi Mengelola Pegawai: A3 *Incomplete Form***

<b>Kode Pengujian</b>	VT-19
<b>Nomor Skenario</b>	Skenario 4
<b>Kode Persyaratan</b>	-
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat memberikan pesan bahwa <i>form</i> pengisian belum terisi secara penuh dan sesuai.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah masuk ke dalam sistem.</li> <li>2. Penguji telah masuk ke dalam halaman tambah pegawai atau sunting pegawai.</li> <li>3. Penguji tidak mengisi <i>form</i> secara penuh dan sesuai.</li> <li>4. Penguji memilih tombol tambah atau sunting (sesuai halaman).</li> </ol>
<b>Hasil yang Diharapkan</b>	Sistem menampilkan pesan bahwa <i>form</i> masih ada salah.
<b>Hasil Pengujian</b>	Sistem menampilkan pesan bahwa <i>form</i> masih ada salah.
<b>Status Uji</b>	Valid

### 7.1.10 Pengujian *Black-Box* Mengelola Pengguna Sistem

Bagian ini menjelaskan pengujian *black-box* pada beberapa fungsi yang dapat digunakan oleh pengguna untuk mengelola pengguna sistem melalui sistem yang telah dikembangkan. Pada Tabel 7.29 menunjukkan beberapa skenario yang bersumber dari alur usecase Mengelola Pengguna Sistem yang akan digunakan untuk mengidentifikasi kasus uji.

**Tabel 7.29 Pengujian *Black-Box* Mengelola Pengguna Sistem**

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	<i>Basic flow</i>	
Skenario 2	<i>Basic flow</i>	A1. Menghapus Pengguna Sistem
Skenario 3	<i>Basic flow</i>	A2. Menyunting Pengguna Sistem
Skenario 4	<i>Basic flow</i>	A3. <i>Incomplete Form</i>

#### 7.1.10.1 Kasus Uji Validasi Mengelola Pengguna Sistem

Pada Tabel 7.30 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-20. Kasus uji VT-20 dilakukan untuk memastikan bahwa sistem dapat menambahkan data pengguna sistem baru dengan kode persyaratan SKPL-F-19. Kasus uji diidentifikasi berdasarkan skenario 1 pada Tabel 7.29. Pengujian *black-box* Mengelola Pengguna Sistem menunjukkan hasil valid.

**Tabel 7.30 Kasus Uji Validasi Mengelola Pengguna Sistem**

<b>Kode Pengujian</b>	VT-20
<b>Nomor Skenario</b>	Skenario 1
<b>Kode Persyaratan</b>	SKPL-F-19
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menampilkan data pengguna sistem yang telah tercatat dan sistem dapat menyimpan data pengguna sistem baru.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah masuk ke dalam sistem.</li> <li>2. Penguji memilih menu pengguna sistem.</li> <li>3. Penguji memilih menu tambah.</li> <li>4. Penguji mengisi <i>form</i> yang telah disediakan dengan benar.</li> <li>5. Penguji memilih tombol tambah.</li> </ol>
<b>Hasil yang Diharapkan</b>	System menampilkan halaman pengguna sistem dengan data pengguna sistem yang tersedia dan sistem dapat menyimpan data pengguna sistem baru yang telah dimasukkan.
<b>Hasil Pengujian</b>	System menampilkan halaman pengguna sistem dengan data pengguna sistem yang tersedia dan sistem dapat menyimpan data pengguna sistem baru yang telah dimasukkan.
<b>Status Uji</b>	Valid

#### 7.1.10.2 Kasus Uji Validasi Mengelola Pengguna Sistem: A1 Menghapus Pengguna Sistem

Pada Tabel 7.31 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-21. Kasus uji VT-21 dilakukan untuk memastikan bahwa sistem dapat menghapus pengguna sistem dengan kode persyaratan SKPL-F-20. Kasus uji diidentifikasi berdasarkan skenario 1 pada Tabel 7.29. Pengujian *black-box* Menghapus Pengguna Sistem menunjukkan hasil valid.

**Tabel 7.31 Kasus Uji Validasi Mengelola Pengguna Sistem: A1 Menghapus Pengguna Sistem**

<b>Kode Pengujian</b>	VT-21
<b>Nomor Skenario</b>	Skenario 2
<b>Kode Persyaratan</b>	SKPL-F-20
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menghapus pengguna sistem yang tersedia.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah masuk ke dalam sistem.</li> <li>2. Penguji memilih menu pengguna sistem.</li> <li>3. Penguji memilih tombol <i>delete</i>.</li> <li>4. Penguji memilih tombol <i>Yes</i> pada pemberitahuan konfirmasi.</li> </ol>
<b>Hasil yang Diharapkan</b>	Sistem menampilkan data pengguna sistem yang tidak dihapus.
<b>Hasil Pengujian</b>	Sistem menampilkan data pengguna sistem yang tidak dihapus.
<b>Status Uji</b>	Valid

#### 7.1.10.3 Kasus Uji Validasi Mengelola Pengguna Sistem: A2 Menyunting Pengguna Sistem

Pada Tabel 7.32 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-22. Kasus uji VT-22 dilakukan untuk memastikan bahwa sistem dapat menyunting data pengguna sistem yang telah tercatat dengan kode persyaratan SKPL-F-21. Kasus uji diidentifikasi berdasarkan skenario 1 pada Tabel 7.29. Pengujian *black-box* Menyunting Pengguna Sistem menunjukkan hasil valid.

**Tabel 7.32 Kasus Uji Validasi Mengelola Pengguna Sistem: A2 Menyunting Pengguna Sistem**

<b>Kode Pengujian</b>	VT-22
<b>Nomor Skenario</b>	Skenario 3
<b>Kode Persyaratan</b>	SKPL-F-21
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menyunting data pengguna sistem yang telah tercatat.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah masuk ke dalam sistem.</li> </ol>



	2. Penguji memilih menu pengguna sistem. 3. Penguji memilih tombol edit. 4. Penguji mengisi <i>form</i> yang telah disediakan. 5. Penguji memilih tombol sunting.
<b>Hasil yang Diharapkan</b>	System menampilkan halaman sunting pengguna sistem, data pengguna sistem dan sistem dapat menyimpan perubahan data pengguna sistem.
<b>Hasil Pengujian</b>	System menampilkan halaman sunting pengguna sistem, data pengguna sistem dan sistem dapat menyimpan perubahan data pengguna sistem.
<b>Status Uji</b>	Valid

#### 7.1.10.4 Kasus Uji Validasi Mengelola Pengguna Sistem: A3 *Incomplete Form*

Pada Tabel 7.33 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-23. Kasus uji VT-23 dilakukan untuk memastikan bahwa sistem dapat menampilkan pesan bahwa *form* belum tersisi secara penuh dan sesuai. Kasus uji diidentifikasi berdasarkan skenario 1 pada Tabel 7.29. Pengujian *black-box Incomplete Form* pada Mengelola Pengguna Sistem menunjukkan hasil -.

**Tabel 7.33 Kasus Uji Validasi Mengelola Pengguna Sistem: A3 *Incomplete Form***

<b>Kode Pengujian</b>	VT-23
<b>Nomor Skenario</b>	Skenario 4
<b>Kode Persyaratan</b>	-
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat memberikan pesan bahwa <i>form</i> pengisian belum terisi secara penuh dan sesuai.
<b>Prosedur Uji</b>	1. Penguji telah masuk ke dalam sistem. 2. Penguji telah masuk ke dalam halaman tambah pengguna sistem atau sunting pengguna sistem. 3. Penguji tidak mengisi <i>form</i> secara penuh dan sesuai. 4. Penguji memilih tombol tambah atau sunting (sesuai halaman).
<b>Hasil yang Diharapkan</b>	Sistem menampilkan pesan bahwa <i>form</i> masih ada salah.
<b>Hasil Pengujian</b>	Sistem menampilkan pesan bahwa <i>form</i> masih ada salah.
<b>Status Uji</b>	Valid

### 7.1.11 Pengujian *Black-Box* Menyunting Profil

Bagian ini menjelaskan pengujian *black-box* pada beberapa fungsi yang dapat digunakan oleh pengguna untuk menyunting profil melalui sistem yang telah dikembangkan. Pada Tabel 7.34 menunjukkan beberapa skenario yang bersumber dari alur *use-case* Menyunting profil yang akan digunakan untuk mengidentifikasi kasus uji.

**Tabel 7.34 Pengujian *Black-Box* Menyunting Profil**

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	<i>Basic flow</i>	
Skenario 2	<i>Basic flow</i>	A1. <i>Incomplete Form</i>

#### 7.1.11.1 Kasus Uji Validasi Menyunting Profil

Pada Tabel 7.35 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-24. Kasus uji VT-24 dilakukan untuk memastikan bahwa sistem dapat menyunting profil pengguna sistem yang telah tercatat dengan kode persyaratan SKPL-F-22. Kasus uji diidentifikasi berdasarkan skenario 1 pada Tabel 7.34. Pengujian *black-box* Menyunting Profil menunjukkan hasil valid.

**Tabel 7.35 Kasus Uji Validasi Menyunting Profil**

<b>Kode Pengujian</b>	VT-24
<b>Nomor Skenario</b>	Skenario 1
<b>Kode Persyaratan</b>	SKPL-F-22
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menyunting profil pengguna sistem.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah masuk ke dalam sistem.</li> <li>2. Penguji memilih menu sunting profil.</li> <li>3. Penguji mengisi <i>form</i> yang telah disediakan.</li> <li>4. Penguji memilih tombol sunting.</li> </ol>
<b>Hasil yang Diharapkan</b>	System menampilkan halaman sunting profil pengguna sistem dan sistem dapat menyimpan perubahan data profil pengguna sistem.
<b>Hasil Pengujian</b>	System menampilkan halaman sunting profil pengguna sistem dan sistem dapat menyimpan perubahan data profil pengguna sistem.
<b>Status Uji</b>	Valid

### 7.1.11.2 Kasus Uji Validasi Menyunting Profil: A1 *Incomplete Form*

Pada Tabel 7.36 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-25. Kasus uji VT-25 dilakukan untuk memastikan bahwa sistem dapat menampilkan pesan bahwa *form* belum tersisi secara penuh dan sesuai. Kasus uji diidentifikasi berdasarkan skenario 1 pada Tabel 7.34 Pengujian *black-box* alternative *Incomplete Form* pada Menyunting Profil menunjukkan hasil valid.

**Tabel 7.36 Kasus Uji Validasi Menyunting Profil: A1 *Incomplete Form***

<b>Kode Pengujian</b>	VT-25
<b>Nomor Skenario</b>	Skenario 2
<b>Kode Persyaratan</b>	-
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat memberikan pesan bahwa <i>form</i> pengisian belum terisi secara penuh dan sesuai.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah masuk ke dalam sistem.</li> <li>2. Penguji telah masuk ke dalam halaman sunting profil pengguna sistem.</li> <li>3. Penguji tidak mengisi <i>form</i> secara penuh dan sesuai.</li> <li>4. Penguji memilih sunting.</li> </ol>
<b>Hasil yang Diharapkan</b>	Sistem menampilkan pesan bahwa <i>form</i> masih ada salah.
<b>Hasil Pengujian</b>	Sistem menampilkan pesan bahwa <i>form</i> masih ada salah.
<b>Status Uji</b>	Valid

### 7.1.12 Pengujian *Black-Box* Mengelola Daftar Tujuan

Bagian ini menjelaskan pengujian *black-box* pada beberapa fungsi yang dapat digunakan oleh pengguna untuk mengelola daftar tujuan melalui sistem yang telah dikembangkan. Pada Tabel 7.37 menunjukkan beberapa skenario yang bersumber dari alur *use-case* Mengelola Daftar Tujuan yang akan digunakan untuk mengidentifikasi kasus uji.

**Tabel 7.37 Pengujian *Black-Box* Mengelola Daftar Tujuan**

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	<i>Basic flow</i>	

No. Skenario	Alur Awal	Alur Alternatif
Skenario 2	<i>Basic flow</i>	A1. Menghapus Daftar Tujuan
Skenario 3	<i>Basic flow</i>	A2. Menyunting Daftar Tujuan
Skenario 4	<i>Basic flow</i>	A3. <i>Incomplete Form</i>

#### 7.1.12.1 Kasus Uji Validasi Mengelola Daftar Tujuan

Pada Tabel 7.38 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-26. Kasus uji VT-26 dilakukan untuk memastikan bahwa sistem dapat menampilkan data daftar tujuan yang telah tercatat dan dapat menambahkan data daftar tujuan baru dengan kode persyaratan SKPL-F-07. Kasus uji diidentifikasi berdasarkan skenario 1 pada Tabel 7.37. Pengujian *black-box* Mengelola Daftar Tujuan menunjukkan hasil valid.

**Tabel 7.38 Kasus Uji Validasi Mengelola Daftar Tujuan**

<b>Kode Pengujian</b>	VT-26
<b>Nomor Skenario</b>	Skenario 1
<b>Kode Persyaratan</b>	SKPL-F-07
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menampilkan data daftar tujuan yang telah tercatat dan sistem dapat menyimpan data daftar tujuan baru.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah masuk ke dalam sistem.</li> <li>2. Penguji memilih menu daftar tujuan.</li> <li>3. Penguji memilih menu tambah.</li> <li>4. Penguji mengisi <i>form</i> yang telah disediakan dengan benar.</li> <li>5. Penguji memilih tombol tambah.</li> </ol>
<b>Hasil yang Diharapkan</b>	System menampilkan halaman daftar tujuan dengan data daftar tujuan yang tersedia dan sistem dapat menyimpan data daftar tujuan baru yang telah dimasukkan.
<b>Hasil Pengujian</b>	System menampilkan halaman daftar tujuan dengan data daftar tujuan yang tersedia dan sistem dapat menyimpan data daftar tujuan baru yang telah dimasukkan.
<b>Status Uji</b>	Valid

#### 7.1.12.2 Kasus Uji Validasi Mengelola Daftar Tujuan: A1 Menghapus Daftar Tujuan

Pada Tabel 7.39 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-27. Kasus uji VT-27 dilakukan untuk memastikan bahwa sistem dapat menghapus daftar tujuan dengan kode persyaratan SKPL-F-08. Kasus uji diidentifikasi berdasarkan skenario 1 pada Tabel 7.37. Pengujian *black-box* Menghapus Daftar Tujuan menunjukkan hasil valid.

**Tabel 7.39 Kasus Uji Validasi Mengelola Daftar Tujuan: A1 Menghapus Daftar Tujuan**

<b>Kode Pengujian</b>	VT-27
<b>Nomor Skenario</b>	Skenario 2
<b>Kode Persyaratan</b>	SKPL-F-08
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menghapus daftar tujuan yang tersedia.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah masuk ke dalam sistem.</li> <li>2. Penguji memilih menu daftar tujuan.</li> <li>3. Penguji memilih tombol <i>delete</i>.</li> <li>4. Penguji memilih tombol <i>Yes</i> pada pemberitahuan konfirmasi.</li> </ol>
<b>Hasil yang Diharapkan</b>	Sistem menampilkan data daftar tujuan yang tidak dihapus.
<b>Hasil Pengujian</b>	Sistem menampilkan data daftar tujuan yang tidak dihapus.
<b>Status Uji</b>	Valid

#### 7.1.12.3 Kasus Uji Validasi Mengelola Daftar Tujuan: A2 Menyunting Daftar Tujuan

Pada Tabel 7.40 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-28. Kasus uji VT-28 dilakukan untuk memastikan bahwa sistem dapat menyunting data daftar tujuan yang telah tercatat dengan kode persyaratan SKPL-F-09. Kasus uji diidentifikasi berdasarkan skenario 1 pada Tabel 7.37. Pengujian *black-box* Menyunting Daftar Tujuan menunjukkan hasil valid.

**Tabel 7.40 Kasus Uji Validasi Mengelola Daftar Tujuan: A2 Menyunting Daftar Tujuan**

<b>Kode Pengujian</b>	VT-28
<b>Nomor Skenario</b>	Skenario 3
<b>Kode Persyaratan</b>	SKPL-F-09
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menyunting data daftar tujuan yang telah tercatat.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah masuk ke dalam sistem.</li> <li>2. Penguji memilih menu daftar tujuan.</li> <li>3. Penguji memilih tombol edit.</li> <li>4. Penguji mengisi <i>form</i> yang telah disediakan.</li> <li>5. Penguji memilih tombol sunting.</li> </ol>
<b>Hasil yang Diharapkan</b>	System menampilkan halaman sunting daftar tujuan dan sistem dapat menyimpan perubahan data daftar tujuan.
<b>Hasil Pengujian</b>	System menampilkan halaman sunting daftar tujuan dan sistem dapat menyimpan perubahan data daftar tujuan.
<b>Status Uji</b>	Valid

#### 7.1.12.4 Kasus Uji Validasi Mengelola Pengguna Sistem: A3 *Incomplete Form*

Pada Tabel 7.41 berisi mengenai informasi rencana pengujian dan kasus uji *black-box* dengan kode VT-29. Kasus uji VT-29 dilakukan untuk memastikan bahwa sistem dapat menampilkan pesan bahwa *form* belum tersisi secara penuh dan sesuai. Kasus uji diidentifikasi berdasarkan skenario 1 pada Tabel 7.37. Pengujian *black-box Incomplete Form* pada Mengelola Daftar Tujuan menunjukkan hasil valid.

**Tabel 7.41 Kasus Uji Validasi Mengelola Pengguna Sistem: A3 *Incomplete Form***

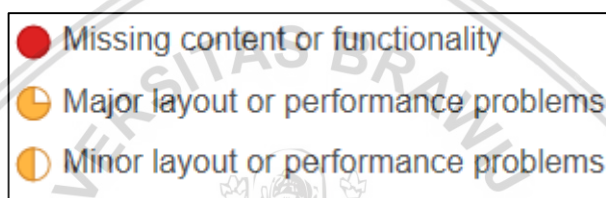
<b>Kode Pengujian</b>	VT-29
<b>Nomor Skenario</b>	Skenario 2
<b>Kode Persyaratan</b>	-
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat memberikan pesan bahwa <i>form</i> pengisian belum terisi secara penuh dan sesuai.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah masuk ke dalam sistem.</li> <li>2. Penguji telah masuk ke dalam halaman tambah daftar tujuan atau sunting daftar tujuan.</li> <li>3. Penguji tidak mengisi <i>form</i> secara penuh dan sesuai.</li> </ol>



	4. Penguji memilih tombol tambah atau sunting (sesuai halaman).
<b>Hasil yang Diharapkan</b>	Sistem menampilkan pesan bahwa <i>form</i> masih ada salah.
<b>Hasil Pengujian</b>	Sistem menampilkan pesan bahwa <i>form</i> masih ada salah.
<b>Status Uji</b>	Valid

## 7.2 Pengujian *Compatibility*

Pengujian dilakukan pada lingkungan implementasi dengan menggunakan aplikasi *SortSite*. Terdapat beberapa kategori skor untuk mengukur tingkat kompatibilitas sesuai dengan yang dijelaskan dalam Gambar 7.2.



**Gambar 7.2 Kategori *compatibility issues* SortSite**

Hasil pengujian kompatibilitas yang terdapat dalam Gambar 7.3 menunjukkan bahwa terdapat *minor issue* pada *browser internet explorer*, *firefox*, dan *blackberry* yang dikarenakan *browser* tersebut tidak mendukung beberapa CSS yang digunakan dalam sistem informasi.

Browser	IE			Edge	Firefox	Safari		Opera	Chrome	iOS			Android	
Version	9	10	11	16	60	≤ 10	11	51	66	≤ 9	10	11	≤ 3	4*
Critical Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Major Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Minor Issues	●	●	●	✓	●	●	✓	✓	✓	✓	✓	✓	✓	✓

**Gambar 7.3 Hasil Pengujian SortSite**

## 7.3 Pengujian *User Acceptance Testing (UAT)*

Pada sub bab ini menjelaskan pengujian menggunakan *User Acceptance Testing* untuk mengetahui sistem informasi yang dikembangkan telah sesuai dengan kebutuhan bisnis yang sudah teridentifikasi pada tahap analisis kebutuhan. Sebelum melakukan *User acceptance testing* terlebih dahulu membuat daftar pernyataan yang mewakili seluruh kebutuhan yang sudah didefinisikan. *User acceptance testing* dilakukan dengan memberikan penjelasan mengenai sistem yang akan diuji untuk memberikan gambaran awal mengenai

sistem yang telah dikembangkan tersebut. Langkah selanjutnya yaitu meminta responden untuk melakukan serangkaian *task* yang telah diberikan. Responden tidak diperkenankan untuk melakukan *task* diluar dari skenario yang diberikan. Responden memberikan penilaian terhadap pernyataan setelah melakukan rangkaian *task* yang diberikan. Berikut Tabel 7.2 sebagai perwakilan pengguna yang akan mengisi daftar pernyataan.

Terdapat beberapa kriteria pada pengujian UAT namun, penulis memilih 3 kriteria untuk pengujian sistem informasi yang telah dikembangkan dalam penelitian ini. Dari ketiga kriteria tersebut antara lain *Performance*, *Usability*, dan *Functional Correctness and Completeness* karena dinilai telah mampu mewakili apakah sistem tersebut dapat diterima oleh pengguna atau tidak. Pengujian ini dilakukan kepada pengguna akhir atau anggota.

**Tabel 7.42 Analisa User Acceptance Testing**

No	Peran	Fungsi Sistem Yang Digunakan	Jumlah Populasi	Jumlah Tester ( $\geq 50\%$ )
1	Pimpinan (Direktur)	1. <i>Login</i>	1	1
		2. Mengelola pengguna sistem		
		3. Mengelola pegawai		
		4. Mencetak laporan unit		
		5. Mencetak laporan global		
		6. Menyunting profil		
		7. <i>Logout</i>		
2	Pimpinan (Komisaris)	1. <i>Login</i>	1	1
		2. Mengelola pengguna sistem		
		3. Mengelola pegawai		
		4. Mencetak laporan unit		
		5. Mencetak laporan global		
		6. Menyunting profil		
		7. <i>Logout</i>		
		8. <i>Login</i>		
3	Bagian Administrasi (Kasir)	1. <i>Login</i>	1	1
		2. Melihat jadwal bus		
		3. Mengelola daftar tujuan		

No	Peran	Fungsi Sistem Yang Digunakan	Jumlah Populasi	Jumlah Tester (>=50%)
		4. Mencetak Slip Gaji		
		5. Menyunting Profil		
		6. Logout		
4	Bagian Administrasi (Pembukuan)	1. Login	1	1
		2. Mengelola armada bus		
		3. Mengelola persewaan		
		4. Mengelola operasional bus		
		5. Mencetak laporan unit		
		6. Mencetak laporan global		
		7. Menyunting profil		
		8. Logout		
5	Bagian Pemasaran	1. Login	4	2
		2. Melihat jadwal bus		
		3. Logout		
Total Tester			8	6

**Tabel 7.43 Kriteria dan Pernyataan User Acceptance Testing**

No	Kriteria	Definisi	Daftar Pernyataan
1.	Usability	Evaluasi layanan atau produk melalui pengujian dengan pengguna yang mewakili yang bertujuan untuk memastikan bahwa sistem bisa mudah digunakan dan dipelajari oleh pengguna.	1. Sistem informasi penggajian karyawan bus pariwisata PT Mahkota Mitra Sentosa mudah digunakan.
			2. Tampilan sistem informasi penggajian karyawan bus pariwisata PT Mahkota Mitra Sentosa nyaman ketika digunakan.
			3. Sistem informasi penggajian karyawan bus pariwisata PT Mahkota Mitra Sentosa mempermudah dalam mencari informasi

No	Kriteria	Definisi	Daftar Pernyataan
			4. Laporan akhir mempermudah dalam melaporkan ke pimpinan
2.	<i>Functional Correctness and Completeness</i>	<i>Functional Correctness and Completeness</i> adalah pengujian untuk mengetahui apakah sistem yang dibuat sudah sesuai kebutuhan yang diinginkan pengguna dan telah didefinisikan pada spesifikasi kebutuhan	<p>1. Sistem informasi penggajian karyawan bus pariwisata PT Mahkota Mitra Sentosa membantu dalam pengelolaan data persewaan dengan baik</p> <p>2. Sistem informasi penggajian karyawan bus pariwisata PT Mahkota Mitra Sentosa membantu dalam pengelolaan data operasional dengan baik</p> <p>3. Sistem informasi penggajian karyawan bus pariwisata PT Mahkota Mitra Sentosa membantu dalam penggajian</p> <p>4. Sistem informasi penggajian karyawan bus pariwisata PT Mahkota Mitra Sentosa membantu dalam pembuatan laporan unit</p> <p>5. Sistem informasi penggajian karyawan bus pariwisata PT Mahkota Mitra Sentosa membantu dalam pembuatan laporan global</p> <p>6. Sistem informasi penggajian karyawan bus pariwisata PT Mahkota Mitra Sentosa membantu dalam kelola armada</p> <p>7. Sistem informasi penggajian karyawan bus pariwisata PT Mahkota Mitra Sentosa dapat diakses 24 jam dalam 7 hari</p>
3.	<i>Confidentially and Availability</i>	<p><i>Confidentiality</i> adalah aturan yang membatasi hak akses terhadap informasi.</p> <p><i>Availability</i> adalah jaminan akses yang dapat dipercaya</p>	1. Diperlukan <i>login user</i> untuk keamanan informasi kantor dalam sistem informasi penggajian karyawan bus pariwisata PT Mahkota Mitra Sentosa

No	Kriteria	Definisi	Daftar Pernyataan
		terhadap informasi oleh orang-orang yang mempunyai wewenang.	

Setelah mendapatkan kriteria dan pernyataan yang sesuai dengan kriteria yang diharapkan. Tahap selanjutnya adalah melakukan analisa hasil dari daftar pernyataan yang sudah dijawab oleh pengguna sistem sebagai *tester*. Kode UAT diwakilkan dengan kode UAT-1, UAT-2, dan seterusnya secara berurutan yang ditunjukkan pada Tabel berikut:

**Tabel 7.1 Hasil User Acceptance Testing**

No.	Pernyataan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1	Sistem informasi penggajian karyawan bus pariwisata PT Mahkota Mitra Sentosa mudah digunakan.	UAT-1				6	
2	Tampilan sistem informasi penggajian karyawan bus pariwisata PT Mahkota Mitra Sentosa nyaman ketika digunakan.	UAT-2			2	4	
3	Sistem informasi penggajian karyawan bus pariwisata PT Mahkota Mitra Sentosa mempermudah dalam mencari informasi	UAT-3					6
4	Laporan akhir mempermudah dalam melaporkan ke pimpinan	UAT-4				4	2

No.	Pernyataan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
5	Sistem informasi penggajian karyawan bus pariwisata PT Mahkota Mitra Sentosa membantu dalam pengelolaan data persewaan dengan baik	UAT-5				4	2
6	Sistem informasi penggajian karyawan bus pariwisata PT Mahkota Mitra Sentosa membantu dalam pengelolaan data operasional dengan baik	UAT-6				6	
7	Sistem informasi penggajian karyawan bus pariwisata PT Mahkota Mitra Sentosa membantu dalam penggajian	UAT-7				6	
8	Sistem informasi penggajian karyawan bus pariwisata PT Mahkota Mitra Sentosa membantu dalam pembuatan laporan unit	UAT-8					6
9	Sistem informasi penggajian karyawan bus pariwisata PT Mahkota Mitra Sentosa membantu dalam pembuatan laporan global	UAT-9					6
10	Sistem informasi penggajian karyawan	UAT-10				6	



No.	Pernyataan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
	bus pariwisata PT Mahkota Mitra Sentosa membantu dalam kelola armada						
11	Sistem informasi penggajian karyawan bus pariwisata PT Mahkota Mitra Sentosa dapat diakses 24 jam dalam 7 hari	UAT-11					6
12	Diperlukan login user untuk keamanan informasi kantor dalam sistem informasi penggajian karyawan bus pariwisata PT Mahkota Mitra Sentosa	UAT-12				6	
Total			0	0	2	42	28

Pada perhitungan skala Likert dari total hasil kuesioner *User Acceptance Testing* untuk pengguna sistem yaitu dengan menghitung nilai berdasarkan dari total responden yang telah dipilih sebagai berikut:

$$\text{Nilai A (sangat setuju)} = 28 \times 5 = 140$$

$$\text{Nilai B (setuju)} = 42 \times 4 = 168$$

$$\text{Nilai C (Netral)} = 2 \times 3 = 6$$

$$\text{Nilai D (tidak setuju)} = 0 \times 2 = 0$$

$$\text{Nilai E (sangat tidak setuju)} = 0 \times 1 = 0$$

$$\text{Total nilai} = \text{Nilai A} + \text{Nilai B} + \text{Nilai C} + \text{Nilai D} + \text{Nilai E}$$

$$\text{Total nilai} = 140 + 168 + 6 + 0 + 0 = 314$$

Setelah menghitung nilai total dari masing-masing nilai, selanjutnya menghitung nilai Y dengan perhitungan sebagai berikut:

$Y = \text{Jumlah Skala} \times \text{Jumlah Responden} \times \text{Jumlah Pernyataan}$

$$Y = 5 \times 6 \times 12 = 360$$

Pada akhir perhitungan menggunakan skala Likert pada *User Acceptance Testing* yaitu dengan menghitung dengan rumus index untuk menghitung besar nilai persentase sebagai berikut:

$$\text{Rumus index} = (\text{Total Nilai} \div Y) \times 100\%$$

$$\text{Maka rumus index} = (314 \div 360) \times 100\% = 87,2\%$$

Dari analisis hasil perhitungan kriteria pada *User Acceptance Testing* menggunakan skala Likert yang telah dilakukan melibatkan enam responden menunjukkan hasil akhir 87,2%. Hasil akhir tersebut dapat diambil kesimpulan bahwa dari responden yang akan menggunakan sistem menyatakan sangat setuju dengan sistem informasi penggajian dapat membantu untuk menjalankan operasionalnya karena sesuai sistem yang telah dibuat sesuai dengan apa yang dibutuhkan.

Pada kriteria *usability* berdasarkan hasil pengujian oleh responden menunjukkan beberapa responden memilih skala sangat setuju atau setuju yang berarti sudah dianggap bagus dalam hal tampilan serta kemudahan menggunakan sistem informasi. Kemudahan menggunakan sistem informasi tersebut dikarenakan responden telah mengetahui dengan jelas fungsi-fungsi dari tombol yang tersedia di sistem dan ukuran tombolnya menurut responden mudah digunakan yang ukurannya tidak terlalu kecil ataupun terlalu besar. Namun, ada juga responden yang memilih netral.

Kemudian, kriteria *Functional Correctness and Completeness* berdasarkan hasil pengujian oleh responden menunjukkan beberapa responden memilih skala sangat setuju atau setuju berarti bahwa sistem yang dibangun telah diterima dengan baik dan sesuai dengan kebutuhan.

Pada kriteria *Confidentially and Availability* berdasarkan hasil pengujian oleh responden menunjukkan semua responden memilih skala setuju. Hasil tersebut menunjukkan bahwa responden memerlukan sistem yang dapat menjamin keamanan dan akses sistem dapat dibatasi untuk setiap pengguna sesuai dengan kebutuhan.

## BAB 8 KESIMPULAN DAN SARAN

### 8.1 Kesimpulan

Berdasarkan dari penelitian yang telah dilakukan, dapat diambil kesimpulan dari pengembangan sistem informasi penggajian karyawan bus pariwisata PT Mahkota Mitra Sentosa menggunakan metode *Rational Unified Process* (RUP) sebagai berikut:

1. Pada fase *inception*, menghasilkan proses bisnis usulan yang dapat mempermudah aktivitas dalam operasional pada PT Mahkota Mitra Sentosa. Hal ini dikarenakan dengan adanya sistem informasi yang dikembangkan dapat membantu menyelesaikan permasalahan yang ada yaitu tidak perlunya menghitung secara manual dalam proses penggajian karena proses tersebut sudah diautomatisasi.
2. Hasil analisis kebutuhan dan perancangan pada fase *elaboration* menghasilkan *vision document*, kebutuhan fungsional dan non fungsional, pemodelan *use-case*, pemodelan aktivitas, *sequence diagram*, *class diagram*, pemodelan data, antarmuka pengguna, dan perancangan algoritme. Kebutuhan fungsional yang telah didefinisikan yaitu sebanyak 22 kebutuhan, sedangkan untuk kebutuhan non fungsional yaitu sebanyak tiga kebutuhan. Pada fase *elaboration* terdapat sebuah iterasi yang menyebabkan adanya perubahan aktor yang akan menggunakan sistem informasi yaitu penambahan aktor bagian pemasaran dan adanya perubahan tampilan pengguna. Bagian pemasaran ini memiliki akses ke sistem informasi untuk mendapatkan jadwal bus. Perubahan tampilan pengguna terjadi pada tampilan laporan unit dan laporan global.
3. Fase *construction*, menghasilkan implementasi dan pengujian sistem informasi. Pada proses implementasi sistem informasi menggunakan *framework codeigniter* untuk membantu dalam proses implementasi berdasarkan rancangan yang telah dibuat. Pengujian sistem informasi menggunakan pengujian *black-box* yang menghasilkan nilai valid sebesar 100%. Selain pengujian tersebut, juga melakukan pengujian *compatibility* dengan bantuan *tool sortsite* dengan hasil yang menunjukkan terdapat *minor issue* pada tiga jenis *browser* dikarenakan *browser* tersebut tidak mendukung beberapa CSS yang digunakan dalam sistem informasi.
4. Pada fase *transisi*, menghasilkan pengujian menggunakan *User Acceptance Testing* (UAT). Pengujian tersebut melibatkan enam responden yang dilakukan dengan memberikan 12 pernyataan dari tiga kriteria uji UAT menunjukkan nilai akhir sebesar 87,2% dari nilai tersebut responden menerima sistem dengan baik dan sistem yang dibuat menunjukkan sesuai dengan kebutuhan.

## 8.2 Saran

Saran yang dapat diberikan sebagai bahan pertimbangan untuk melaksanakan pengembangan lanjut pada sistem informasi penggajian karyawan bus pariwisata PT Mahkota Mitra Sentosa sebagai berikut:

1. Sistem dapat dikembangkan dengan penambahan fitur yang dapat mengelola penggajian semua karyawan pada PT Mahkota Mitra Sentosa sehingga penggajian semua karyawan tidak hanya sopir dan kernet dapat dikelola pada sistem informasi.
2. Sistem dapat dikembangkan dengan penambahan fitur *tracking GPS* pada setiap bus yang digunakan untuk memantau perjalanan bus.



## DAFTAR PUSTAKA

- (BPMN), B. P. M. a. N., 2011. *Object Management Group, Inc. (OMG)*. [Online] Available at: <http://www.omg.org/spec/BPMN/2.0/PDF/> [Diakses 18 January 2018].
- Becker, et al., 2013. *Process Management: A Guide for the Design of Business Processes*, Berlin: Springer-Verlag.
- Bittner, K. & Spence, I., 2002. *Use-Case Modeling*. Dalam: s.l.:Addison Wesley, p. 368.
- Choizes, 2018. *Pengertian Skala Likert dan Contoh Cara Hitung Kuesionernya*. [Online] Available at: <https://www.diedit.com/skala-likert/> [Diakses 10 Mei 2018].
- DSDM, 2014. *MoSCoW Prioritisation*. [Online] Available at: <https://www.agilebusiness.org/content/moscow-prioritisation> [Diakses 16 Februari 2018].
- Hasibuan, M. . S., 2010. *Manajemen Sumber Daya Manusia*. Jakarta: Bumi Aksara.
- IBM, 1998. *Rational Unified Process, Best Practices for Software*. Cupertino: Rational Software.
- IBM, 2004. *Mastering Object-Oriented Analysis and Design with UML 2.0*. Cupertino: IBM Rational Software.
- Jayanti, D. & Iriani, S., 2014. Sistem Informasi Penggajian Pada CV. Blumbang Sejati Pacitan. *Sentra Penelitian Engineering dan Edukasi*, Volume 6, pp. 36-43.
- Jin, L. & Liang, X., 2012. *System Modeling of Vehicle Management Based on RUP and UML*. Hangzhou, IEEE, pp. 53-56.
- Kadir, A., 2013. *Pengenalan Sistem Informasi*. Edisi Revisi penyunt. Yogyakarta: Andi.
- Kinasih, A. D., Amalia, F. & Priyambadha, B., 2018. Pengembangan Sistem Evaluasi Pembelajaran PAUD (Studi Kasus di PAUD Seruni 05 Kota Malang). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 2, pp. 1027-1035.
- Kroenke, D. M., 2008. *Experiencing MIS*. Upper Saddle River, N.J: Pearson Prentice Hall.
- Monk, E. F. & Wagner, B. J., 2013. *Concepts in Enterprise Resource Planning*. Boston: Course Technology.
- Naik, K. & Tripathy, P., 2008. *Software Testing and Quality Assurance: Theory and Practice*. Dalam: New Jersey: John Wiley & Sons, Inc., Hoboken.
- Nugroho, A., 2010. *Rekayasa Perangkat Lunak Berbasis Objek dengan Metode USDP*. Yogyakarta: Andi.

O'Brien, J. A. & Marakas, G. M., 2010. *Management Information Systems*. Eight Edition penyunt. New York: McGraw-Hill/Irwin.

Pressman, R. S., 2010. *Software Engineering : A Practitioner's Approach*. Dalam: 7th penyunt. New York: McGraw-Hill.

Przybyłek, A., 2011. *Bridging the gap between business process models and use-case models*, Gdańsk: Gdańsk University of Technology.

Quadri, S. M. & Farooq, S. U., 2010. Software Testing - Goals, Principles, and Limitations. *International Journal of Computer Applications*, September, Volume 6, pp. 7-11.

Rahayu, S. & Subagja, M. F. H., 2017. Perancangan Aplikasi Penggajian Berbasis Web Di Sekolah Menengah Kejuruan Islam Atturmudziyyah Garut. *Jurnal Algoritma Sekolah Tinggi Teknologi Garut*, Volume 14, pp. 284-293.

Rizki, I., P., R. A. S. & Yuniati, Y., 2014. Pembuatan Sistem Informasi Penjualan Berbasis Web (Studi Kasus pada Toko Ali Computer). *Jurnal Rekayasa dan Teknologi Elektro*, 8(1), pp. 37-44.

Sommerville, I., 2011. *Software Engineering*. Dalam: 9nd penyunt. Boston: Addison-Wesley.

Valacich, J. & Schneider, C., 2012. *Information System Today Managing in the Digital World*. 5th penyunt. New York: Prentice Hall.

